GYRO: ANCHORING REPRESENTATIONS TO SIMPLEX FRAMES FOR CONTINUAL SELF-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Geometric self-supervised learning uses spectral objectives as a principled foundation for extracting robust object manifolds from unlabeled data streams. However, when applied in a continual setting, newly learned representations rotate freely and drift into previously established object manifolds. Here, we introduce GYRO, a continual self-supervised learning method that minimizes this drift by anchoring representations to the nearest persistent simplex frames. Building on Manifold Capacity Theory, GYRO constrains rotational invariance and enforces geometric stability without requiring class labels, auxiliary alignment losses, or rigid image-to-prototype assignments. Our approach acts directly on the geometry of the embedding space, allowing the network to flexibly adapt its parameters while maintaining separable object manifolds. Extensive evaluations across standard benchmarks show that GYRO consistently outperforms replay, distillation, and regularization baselines. Our results highlight that injecting persistent geometric references transforms unconstrained spectral objectives into stable continual learners, suggesting that much of representation drift is a problem of orientation rather than of network capacity.

1 INTRODUCTION

Learning from unlabeled data streams is central to building systems that adapt over time. Self-supervised learning offers a path to robust representations without human annotation, and now matches or exceeds supervised pretraining on many benchmarks (Chen et al., 2020; Grill et al., 2020; Zbontar et al., 2021; Bardes et al., 2022). Methods grounded in manifold capacity theory (Chung et al., 2018; Cohen et al., 2020) give this learning an explicit geometry. Maximum Manifold Capacity Representations (MMCR) maximize the nuclear norm of the matrix of centroids of augmented views on a hypersphere, which pushes object manifolds to be compact and mutually separable (Yerxa et al., 2023; Schaeffer et al., 2024). This yields strong representations in the i.i.d. setting, yet the geometric stability of such unconstrained representations under the sequential updates of continual learning has not been examined.

In self-supervised continual learning, stability is lost through representation drift, where features learned on new tasks settle into subspaces that overlap with those of earlier tasks (French, 1999; Madaan et al., 2022). Standard remedies act elsewhere: parameter regularization constrains weights (Kirkpatrick et al., 2017; Zenke et al., 2017), distillation constrains outputs against a frozen teacher (Li & Hoiem, 2016; Fini et al., 2022), and replay revisits stored samples (Rebuffi et al., 2017; Buzzega et al., 2020; Cha et al., 2021). None of them constrains the geometry of the embedding space directly. This matters because the nuclear norm depends only on singular values and is therefore invariant to global rotations. The objective therefore keeps no reference that persists across batches: earlier tasks count only through the few replayed images in the current batch, and only relative to the other rows of that batch, never relative to where they were when learned. Nothing ties the region an earlier task occupies to the region it occupied before, which makes drift in large part a problem of orientation (Figure 1).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

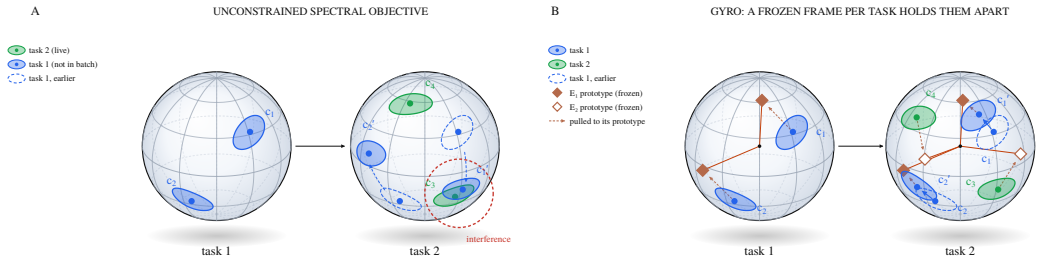

Figure 1: **Persistent frames constrain drift across tasks.** (A) The nuclear-norm objective is invariant to global rotations and provides no persistent directional reference. Earlier manifolds (blue; dashed outlines show prior positions) can drift and overlap with new manifolds (green). (B) GYRO freezes a simplex frame for each task and mixes each centroid with its nearest prototype before evaluating the nuclear norm

To address this, we introduce GYRO, a continual self-supervised learning method that reduces rotational invariance by anchoring representations to persistent reference directions. GYRO mixes each image centroid with its nearest task-specific simplex prototype before the spectral objective is evaluated, so fixed references constrain the objective itself rather than acting as a penalty beside it. Prototype methods typically obtain uniformity from Sinkhorn balancing or centering (Caron et al., 2020; 2021) or from class supervision as in neural collapse (Papyan et al., 2020). GYRO needs neither: it uses no labels, balanced assignments, or alignment loss, and the nuclear norm alone prevents collapse. The anchor enters each centroid as a weighted extra view at a fixed location, and an exact identity shows how anchoring re-weights the MMCR deficit: the view-spread term shrinks, an anchor-distortion term appears, and the spectral term is computed on the mixed centroids. Across standard benchmarks, GYRO outperforms replay, distillation, and regularization baselines and keeps separated object manifolds intact during continual, sequential task learning.

Our main contributions are:

- We study continual self-supervised learning with a manifold capacity objective and identify the lack of a persistent reference, formally the rotational invariance of the nuclear norm, as a source of representation drift.
- We introduce GYRO, which mixes image centroids with persistent, task-specific simplex prototypes inside the spectral objective, without labels, balanced assignment, or an alignment loss.
- We characterize which rotations remain free under the anchored objective, and extend the MMCR deficit identity of Schaeffer et al. (2024) to show exactly how anchoring re-weights view spread, adds an anchor-distortion term, and changes the spectral term.
- GYRO outperforms replay, distillation, and regularization baselines, works alongside replay, needs only a few prototypes, and ablations show that anchoring acts through replay and requires a separate frame per task.

2 Related Works

Self-supervised learning. Joint-embedding methods learn invariance to augmentations and differ mainly in how they prevent collapse: contrastive methods repel negatives (van den Oord et al., 2018; Chen et al., 2020; He et al., 2020), clustering methods predict prototype assignments (Caron et al., 2018; 2020), distillation methods rely on asymmetric architectures (Grill et al., 2020; Chen & He, 2021; Caron et al., 2021), and redundancy-reduction methods de-correlate embedding dimensions (Zbontar et al., 2021; Bardes et al., 2022) while masked modeling offers a reconstructive alternative (He et al., 2022). Maximum Manifold Capacity Representations(MMCR) (Yerxa et al., 2023) instead derives from manifold capacity theory (Chung et al., 2018): maximizing the nuclear norm of per-image view centroids both compresses each image's view manifold and spreads centroids apart, with a single term and no negatives, prototypes, predictors, or momentum encoders. It

matches or exceeds leading joint-embedding methods and admits a clean geometric analysis (Yerxa et al., 2023; Schaeffer et al., 2024). Its explicit centroids make it natural to insert fixed reference directions as pseudo-views, and its invariance to global rotations, harmless in i.i.d. training, leaves representational orientation unconstrained across tasks.

Continual learning. Continual learning methods regularize important parameters or past outputs (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018; Li & Hoiem, 2016), replay stored or generated data (Chaudhry et al., 2019b; Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019a; Rebuffi et al., 2017; Buzzega et al., 2020; Shin et al., 2017), or allocate task-specific capacity (Rusu et al., 2016; Mallya & Lazebnik, 2018; Serrà et al., 2018). Continual self-supervised methods mostly distill from a frozen previous encoder (Gomez-Villa et al., 2022; Fini et al., 2022; Cha & Moon, 2023; Gomez-Villa et al., 2024; Cignoni et al., 2025) or combine the SSL loss with replay (Madaan et al., 2022; Zhang et al., 2024). Closest to `GYRO` are prototype-based methods: iCaRL and PASS store class means or prototypes (Rebuffi et al., 2017; Zhu et al., 2021), and neural-collapse-inspired methods (Papyan et al., 2020; Yang et al., 2022) align features to fixed simplex ETF vertices assigned per class (Yang et al., 2023; Seo et al., 2024; Dang et al., 2025; Wang et al., 2026). `GYRO` differs in using per-task frames with K unrelated to class count, label-free assignments recomputed at every step, and no alignment or classification loss; prototypes act as geometric references, not class targets.

Our position. `GYRO` pairs MMCR's single spectral objective with persistent fixed geometry, without labels, a frozen teacher, or auxiliary losses. Anchors enter the objective as fixed pseudo-views inside each centroid, so view variance, anchor mismatch, and spectral imbalance remain coupled through one nuclear norm, and $\beta = 0$ recovers MMCR exactly. Fixing the anchors reduces MMCR's rotational symmetry and, together with replay, makes cross-task reorientation costly at $O(TKD)$ storage (Proposition 1). `GYRO` is task-aware and replay-based but class-blind; we evaluate its effect on forgetting empirically rather than claiming it by construction. These results are addressed in Section 5.

3 PRELIMINARIES

Setting. We study continual self-supervised learning, in which a model learns from a sequence of unlabeled tasks $\mathcal{D}_1, \ldots, \mathcal{D}_T$, adapting to each new task with the goal of retaining useful representations of earlier data. Task boundaries are known, and a reservoir $\mathcal{M}$ holds a limited set of images from earlier tasks for replay. During task t, each training batch contains n_{cur} images from $\mathcal{D}_t$ and n_{rep} replayed images, for $P = n_{\text{cur}} + n_{\text{rep}}$ images in total; replay begins with the second task, so $n_{\text{rep}} = 0$ on the first. Each stored image keeps the index s_i of its task of origin, and current images have $s_i = t$. Class labels are used only to evaluate the learned representation. The network f_θ consists of a backbone and a D-dimensional projector; the loss is computed on projector outputs, and evaluation uses backbone features.

Maximum manifold capacity representations. MMCR (Yerxa et al., 2023) targets two complementary properties: augmented views of the same image map to similar embeddings, and different images spread across diverse directions of the embedding space. A single objective on per-image centroids encourages both. For each image x_i, the embeddings of its V augmented views $u_{iv}(x_i)$ are ℓ_2-normalized and averaged to form the image's centroid:

$$z_{iv} = \frac{f_\theta(u_{iv}(x_i))}{\|f_\theta(u_{iv}(x_i))\|_2}, \qquad c_i = \frac{1}{V}\sum_{v=1}^{V} z_{iv}. \tag{1}$$

The centroid is used without renormalization. Because each view embedding has unit norm, the norm of the centroid measures how closely the views agree: it approaches one when the views point in similar directions and decreases as they spread apart. Precisely,

$$v_i := \frac{1}{V}\sum_{v=1}^{V} \|z_{iv} - c_i\|_2^2 = 1 - \|c_i\|_2^2. \tag{2}$$

Stacking the centroids as the rows of $C \in \mathbb{R}^{P\times D}$ (all vectors are rows), MMCR minimizes

$$\mathcal{L}_{\text{MMCR}} = -\|C\|_*, \qquad \|C\|_* = \sum_{j=1}^{r} \sigma_j(C), \qquad r = \min(P, D). \tag{3}$$

The nuclear norm rewards centroids of larger norm and, for a fixed sum of squared norms, is largest when the centroids spread evenly over r directions. A single term therefore rewards consistency within each image's views and diversity across images.

The MMCR objective depends on C only through its singular values, so a common orthogonal transformation leaves it unchanged: $\|CR\|_* = \|C\|_*$ for every $R \in O(D)$. Such a transformation preserves pairwise distances and therefore retains the relative geometry of the representation. In continual learning, updates to the shared network can affect earlier and current images differently, changing their relative positions. Replay allows MMCR to organize a sample of earlier images together with current ones, using embeddings computed by the current network. Its objective thus evaluates the geometry of the sampled batch, while continuity with the geometry established during earlier tasks remains implicit. This distinction motivates persistent reference directions, which provide a common geometric reference across successive updates. Section 4 introduces a fixed frame for each task and incorporates these references into the MMCR objective through current and replayed images.

4 METHOD

GYRO extends MMCR with fixed reference directions. Each task receives a randomly oriented simplex frame that remains fixed after initialization. Every time an image is presented, its centroid is mixed with the nearest prototype of its task's frame, and the nuclear norm is evaluated on the mixed centroids. Appendix A provides the training algorithm (Algorithm 1).

4.1 PERSISTENT TASK-SPECIFIC FRAMES

For each task t, GYRO introduces a persistent geometric reference: a frame $E_t \in \mathbb{R}^{K \times D}$ whose K rows, which we call *prototypes*, are unit vectors forming the vertices of a regular simplex centered at the origin. The prototypes have pairwise inner products $-1/(K-1)$, sum to zero, and span a $(K-1)$-dimensional subspace, an arrangement known as a simplex equiangular tight frame (Papyan et al., 2020). The number of prototypes satisfies $2 \le K \le D+1$, the range in which such a simplex fits in the embedding space. GYRO chooses K independently of the class count, so the prototypes serve as geometric reference directions.

Each frame is randomly oriented, independently across tasks, and fixed for the rest of training, so the bank $\{E_1, \dots, E_t\}$ is a collection of task-level simplices rather than a single global ETF. Appendix A.1 gives the construction and its geometric properties.

Each image uses the frame of its task of origin: current images select anchors from E_t, and replayed images use the frame E_{s_i} given by their stored task index (Section 3).

4.2 ANCHORED NUCLEAR NORM

Each image selects an anchor from its task's frame using its current centroid c_i (Equation (1)). With unit-norm prototypes, the nearest prototype in Euclidean distance is the one with the largest inner product:

$$k_i = \operatorname*{arg\,max}_{k \in \{1,\dots,K\}} \big\langle \mathrm{sg}(c_i), e_{s_i,k} \big\rangle, \qquad a_i = e_{s_i,k_i}, \tag{4}$$

where $e_{s_i,k}$ is row k of E_{s_i} and sg denotes the stop-gradient. Assignments are recomputed whenever an image is presented, and several images may select the same prototype.

Each centroid is combined with its selected anchor using a fixed mixing weight $\beta \in [0, 1)$, shared by current and replayed images:

$$y_i = (1-\beta)\, c_i + \beta\, a_i, \qquad \mathcal{L}_{\mathrm{GYRO}}(\theta) = -\|Y\|_* = -\big\|(1-\beta)C + \beta A\big\|_*, \tag{5}$$

where C, A and Y stack the centroids, selected anchors and mixed centroids as rows. The weight β sets the anchor's contribution to each mixed centroid: $\beta = 0$ recovers MMCR on the same batch, and $\beta < 1$ keeps a gradient path to the embeddings. Gradients propagate through current and replayed embeddings; the anchors and their assignment are constants, and the anchor selected for an image can change as its representation evolves.

4.3 Geometry of the Anchored Objective

An exact decomposition. Every centroid has norm at most one, so $\|C\|_* \le \sqrt{Pr}$ (Schaeffer et al., 2024, Prop. 3.3). We call $1 - \|C\|_*^2/(Pr)$ the *deficit*: the normalized shortfall of the squared nuclear norm from this bound. Let $\bar{v} = \frac{1}{P}\sum_i v_i$ be the mean view spread (Equation (2)) and, for any $P \times D$ matrix M, let $\Delta(M) = r\|M\|_F^2 - \|M\|_*^2$ be its spectral imbalance, which is nonnegative by Cauchy–Schwarz and equals zero exactly when all r singular values, zeros included, are equal. With $\|C\|_F^2 = P(1-\bar{v})$, the MMCR deficit splits exactly as

$$1 - \frac{\|C\|_*^2}{Pr} = \bar{v} + \frac{\Delta(C)}{Pr}. \tag{6}$$

The first term measures disagreement between views, the second uneven use of directions, and the bound is attained exactly when both are zero.

Substituting Equation (1) into Equation (5) gives $y_i = \sum_v \frac{1-\beta}{V} z_{iv} + \beta a_i$, so the anchor acts as an additional view with weight β: a fixed prototype rather than an augmentation of the image. Let $\nu_i = \langle c_i, a_i \rangle$ be the alignment of c_i with its anchor, the largest inner product with a prototype of its frame; the prototypes sum to zero, so $\nu_i \ge 0$. Let $\bar{d} = \frac{1}{P}\sum_{i=1}^{P} \|c_i - a_i\|_2^2$ be the mean anchor distortion, and define

$$q_{\text{view}} = (1-\beta)\,\bar{v}, \qquad q_{\text{anchor}} = \beta(1-\beta)\,\bar{d}, \qquad q_{\text{spec}} = \frac{\Delta(Y)}{Pr}. \tag{7}$$

Theorem 1 (Anchored objective)**.** *Fix a batch whose task frames have all been initialized, $\beta \in [0,1)$, and $m = \min\{P,\, D,\, K - 1 + n_{\text{rep}}\}$.*

(i) Decomposition. *For each row,*

$$1 - \|y_i\|_2^2 = (1-\beta)\, v_i + \beta(1-\beta)\, \|c_i - a_i\|_2^2,$$

the weighted spread around y_i of the V views, each with weight $\frac{1-\beta}{V}$, and the anchor, with weight β. Averaging over the rows gives, exactly,

$$1 - \frac{\|Y\|_*^2}{Pr} = q_{\text{view}} + q_{\text{anchor}} + q_{\text{spec}}. \tag{8}$$

(ii) Exact prototype matching. $\operatorname{rank} A \le m$ *for every assignment. If $c_i = a_i$ for every row, then $\|Y\|_* = \|A\|_* \le \sqrt{Pm}$. If $P \le D$, every C with orthonormal rows gives $\|Y\|_* \ge (1-\beta)P$. Hence, when $P \le D$ and $(1-\beta)^2 > m/P$, exact prototype matching is not a global maximizer of $\|Y\|_*$ over centroid matrices with rows in the unit ball.*

The proof is in Appendix A. At $\beta = 0$, Equation (8) reduces to Equation (6). For fixed embeddings, anchoring scales the view term by $1-\beta$, adds the anchor term, and evaluates the spectral imbalance on Y instead of C. All three terms depend on the same mixed centroids, so they describe one coupled objective rather than three separately weighted losses.

Earlier tasks enter through replay. Each image selects its anchor from its own task's frame, so the loss depends only on the frames of the tasks in the batch, $\mathcal{T}_B = \{t\} \cup \{s_i : i \text{ replayed}\}$. Replayed images bring the frames of earlier tasks into the current loss; without replay, the loss at task t depends on E_t alone. Orthogonal transformations that preserve these frames leave the loss unchanged, and in expectation over reservoir draws the guaranteed set of such transformations becomes smaller as retained tasks accumulate (Proposition 1).

Task-specific frames. Since $\|c_i - a_i\|_2^2 = 1 + \|c_i\|_2^2 - 2\nu_i$, the anchor distortion of a replayed image increases when drift lowers its alignment with every prototype of its original frame, and it is largest when c_i is orthogonal to that frame (Appendix A.2). With one frame shared by all tasks, movement between its prototypes can remain inexpensive, so each task receives its own reference directions (Appendix B).

Prototypes remain references. Exact prototype matching, $c_i = a_i$ for every row, sets the anchor distortion to zero but confines the batch to the span of the anchors, of rank at most m; for one frame, the realized rank is set by prototype occupancy (Lemma 1). By Theorem 1(ii), exact matching is not a global maximizer when $P \le D$ and $\beta < 1 - \sqrt{m/P}$. For the CIFAR-100 batches this covers $\beta = 1/3$ for $K \le 10$ (Appendix A).

Choosing β. For fixed centroids and anchors, $\|Y\|_*$ is convex in β, and when $\|A\|_* < \|C\|_*$ every $\beta \in (0, 1)$ gives a smaller value than $\beta = 0$ (Proposition 2). The intended benefit of anchoring concerns retention during later tasks, so we select β by continual-learning performance (Section 6.2). Changing β changes the direction of the gradient as well as its scale; rescaling the learning rate by $1/(1-\beta)$ changes accuracy by at most 0.5 points (Appendix B).

5 EXPERIMENTS

5.1 SETUP

Benchmarks. We evaluate on two natural-image benchmarks, each partitioned into class-incremental tasks. Split CIFAR-100 divides 100 classes at three granularities: 5 tasks of 20 classes, 20 tasks of 5 classes, and 50 tasks of 2 classes. Split Tiny-ImageNet divides 200 classes into 20 tasks of 10 classes, 50 tasks of 4 classes, and 100 tasks of 2 classes. Both benchmarks use the same training augmentations, differing only in crop size, and evaluation uses the normalized image with no augmentation. The class stream is contiguous in label order and identical across seeds, so a seed changes only the initialization, the augmentation randomness, the batch order and the replay draws. Dataset sizes and augmentations are given in Table S2.

Implementation. A ResNet-50 backbone with batch normalization, a projector, and ℓ_2-normalized embeddings is trained from scratch with Adam. Evaluation reads the backbone feature from a linear probe fit on the frozen backbone after each task boundary (Fini et al., 2022). Replay uses class-blind reservoir sampling, updated at the end of each task. The first epoch of the first task is unanchored, and every later task is anchored from its first step. Remaining settings are given in Table S2 and Appendix D.

Baselines. We compare against plain sequential fine-tuning (Fine-tune), which optimizes the MMCR objective with no continual mechanism, and a jointly-trained upper bound, which sees all data at once. The continual baselines are weight regularization (EWC (Kirkpatrick et al., 2017)), distillation (LwF (Li & Hoiem, 2016), CaSSLe (Fini et al., 2022)), gradient projection (GPM (Saha et al., 2021)) and null-space gradient projection (NGP), replay (ER (Chaudhry et al., 2019b), DER (Buzzega et al., 2020), DER++ and LUMP (Madaan et al., 2022)), and ParamProj, a null-space projection on weights of our own design following (Nayak et al., 2025). All methods are adapted to the MMCR self-supervised loss; the loss formulation used for each is given in Appendix D. Each method's own hyperparameter was swept and the best setting selected per task granularity; the selected values are given in Table S3. In Table 1 every method that replays does so at the same budget. Table S7 reports what larger budgets buy.

Evaluation Protocols. We report two metrics in Tables 1 and 2, both read from a linear probe fit on the frozen backbone after each task boundary. **Final** is Class-IL accuracy after the last task, from one probe trained over every class seen so far. **Forg** is average forgetting, the drop for each task from the best accuracy it ever reached to its accuracy after the last task. Task-IL average accuracy and backward transfer are reported in Appendix E, and the formula for each metric is given in Table S8.

5.2 RESULTS

Table 1 compares the methods on Split CIFAR-100 across three task granularities. Fine-tuning remains competitive on the five-task sequence, where most regularization and distillation methods offer limited improvements. LUMP performs best in this setting, followed by `GYRO`. As the sequence becomes longer, replay becomes increasingly beneficial, and `GYRO` achieves the highest final accuracy on both the 20- and 50-task sequences. Its advantage over LUMP, the strongest competing baseline, increases from 1.0 to 2.0 percentage points, suggesting that persistent reference directions are most useful when learning proceeds through many tasks with few classes each.

`GYRO` 's accuracy gains on the longer sequences are accompanied by the lowest forgetting. However, retention alone does not explain the performance ranking. CaSSLe exhibits relatively low forgetting on the 50-task sequence but lower final accuracy than `GYRO`. Conversely, LUMP outperforms ER despite slightly greater forgetting on both longer sequences.

Table 1: Results on Split CIFAR-100, 20 epochs per task, ResNet-50. Each method is shown at the best setting of its own sweep. Mean±std over three seeds. The best model in each column is made bold. We separate replay-based (bottom) and other methods (top) for easier comparisons. Further metrics and settings in Tables S5 and S7.

	5 tasks		20 tasks		50 tasks	
Method	**Final↑**	**Forg↓**	**Final↑**	**Forg↓**	**Final↑**	**Forg↓**
Fine-tune	61.0±0.4	2.3±0.2	54.4±0.5	3.1±1.0	50.0±0.7‡	3.3±0.3‡
EWC	61.5±0.4	2.0±0.1	54.3±0.4	3.1±0.5	49.6±0.7	4.0±0.9
CaSSLe	61.4±0.6	2.3±0.3	54.3±0.6	3.2±0.3	51.1±0.8	1.6±0.2
NGP	61.3±0.4	1.8±0.1	54.6±0.9	3.5±0.1	50.2±0.3	3.5±0.4
Param. Proj.	61.5±0.5	2.1±0.3	54.2±0.3	2.9±0.0	49.8±0.2	2.6±0.3
ER	60.9±0.4	2.0±0.3	56.6±0.7	2.2±0.1	53.6±0.2	1.9±0.2
DER	60.4±0.4	2.5±0.9	54.0±0.6	3.5±0.2	50.1±0.4	2.9±0.2
DER++	60.3±0.1	1.9±0.4	55.9±0.4	2.7±0.5	54.0±0.2	2.3±0.4
LUMP	**63.2±0.3**	**1.3±0.3**	58.3±0.6	2.4±0.4	55.2±0.1	2.2±0.6
GYRO (Ours)	61.7±0.3	1.5±0.3	**59.3±0.5**	**1.2±0.1**	**57.2±0.5**	**1.2±0.1**
Joint (UB)	66.3±0.7	0.0	66.3±0.7	0.0	66.3±0.7	0.0

Table 2: Split Tiny ImageNet, 20 epochs per task, ResNet-50, batch 32 with 40 views per image. Each method is shown at the best setting of its own sweep (Section G). Mean±std over three seeds. **Joint** sees all data at once and is not directly comparable. Further metrics in Table S6.

	20 tasks		50 tasks		100 tasks	
Method	**Final↑**	**Forg↓**	**Final↑**	**Forg↓**	**Final↑**	**Forg↓**
Fine-tune	26.0±0.8	9.9±0.4	25.1±0.7	7.1±0.6	25.4±0.6	7.1±0.4
EWC	30.1±0.7	8.9±0.5	27.7±0.3	7.8±0.3	24.6±0.1	7.6±0.3
CaSSLe	26.6±0.4	9.2±0.4	26.1±1.0	6.4±0.9	25.2±0.5	7.2±0.1
NGP	26.4±0.4	9.6±1.1	27.2±0.6	6.7±0.2	25.5±0.4	6.3±0.2
Param. Proj.	25.9±0.7	10.0±0.5	25.1±0.4	6.6±0.3	23.7±0.6	5.2±0.1
ER	20.8±1.1	10.9±0.2	20.3±0.6	7.8±0.2	20.0±0.7	5.7±0.1
DER	26.3±0.4	9.2±0.7	26.3±0.6	8.2±0.5	25.0±0.5	6.9±0.4
DER++	21.7±1.4	10.7±0.7	21.3±0.5	8.3±0.2	19.9±0.6	5.8±0.7
LUMP	28.8±0.2	8.6±0.7	27.4±0.7	6.7±0.7	27.5±1.0	4.7±0.2
GYRO (Ours)	**30.2±1.9**	**5.0±0.6**	**31.3±0.9**	**3.3±0.3**	**31.3±0.9**	**4.4±0.1**
Joint (UB)	34.66±0.76	0.0	34.66±0.76	0.0	34.66±0.76	0.0

Table 2 extends the comparison to Split Tiny ImageNet, where GYRO achieves the highest mean final accuracy and lowest mean forgetting across all three task granularities. On the 20-task sequence, its accuracy is comparable to EWC within the reported seed variation, while its forgetting is considerably lower. The advantage becomes clearer on the longer sequences: GYRO exceeds the strongest competing baselines, EWC and LUMP, by 3.6 and 3.8 percentage points on the 50- and 100-task sequences, respectively.

Unlike on CIFAR-100, replay alone underperforms fine-tuning on Tiny ImageNet under the evaluated configuration. DER++ follows a similar pattern, whereas GYRO and LUMP both improve over fine-tuning. Fine-tune is therefore a reference point rather than a lower bound: several continual methods fall below it in both benchmarks, and on the longest Tiny ImageNet sequence most of them do.

Finally, GYRO maintains its mean final accuracy as the Tiny ImageNet sequence grows from 50 to 100 tasks.

6 ANALYSIS

We first measure what anchoring preserves (Section 6.1), and then ask how much anchoring is needed and whether the frame size matters (Section 6.2).

6.1 ANCHORING PRESERVES PREVIOUSLY LEARNED GEOMETRY

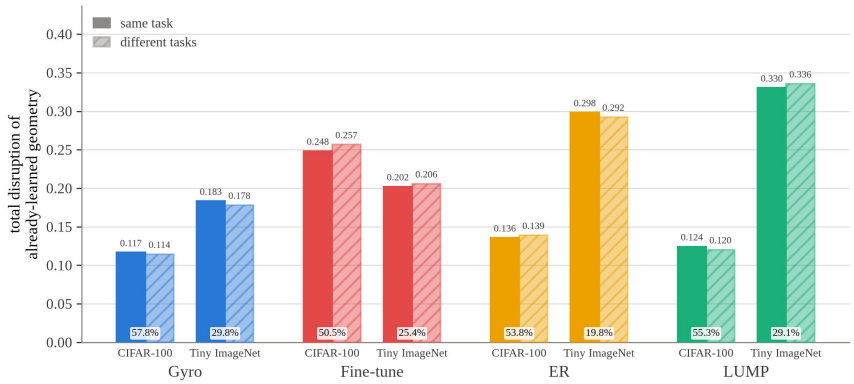

Figure 2: **Preservation of learned geometry.** Accumulated mean squared change in pairwise cosine similarity between backbone features of previously seen images, summed over all task boundaries, for pairs from the same task (solid bars) and from different tasks (lighter hatched bars), on Split CIFAR-100 (50 tasks) and Split Tiny ImageNet (100 tasks). The value at the base of each pair is the final Class-IL accuracy of the same run. ResNet-50, one run per method; on Tiny ImageNet, the best of three seeds, with ER on a different seed. Details in Appendix F.1.

At each task boundary, we accumulate the mean squared change in pairwise cosine similarity between backbone features of previously seen images, the representation used for linear evaluation, separately within and across tasks. The measure ignores common rotations, which the unanchored objective leaves free and the anchored one constrains (Proposition 1), so it compares changes in relative geometry rather than orientation.

On both benchmarks, GYRO accumulates the lowest disruption (Figure 2). On CIFAR-100, its $\beta = 0$ twin, which differs only in the mixing weight and is not shown, accumulates 0.164 within and 0.153 across tasks; relative to it, GYRO reduces disruption by 29% and 25%, and average forgetting from 2.62 to 1.26 percentage points. The only term the merge adds is the anchor distortion (Theorem 1(i)), which grows as drift lowers a replayed image's alignment with its own task's frame (Section 4.3), consistent with this reduction. Across the four methods shown, lower disruption also accompanies higher final accuracy. On Tiny ImageNet, disruption is 39% lower than for ER and 45 to 47% lower than for LUMP; this comparison complements the controlled CIFAR-100 twin experiment.

6.2 JOINT SENSITIVITY TO ANCHORING STRENGTH AND FRAME SIZE

Figure 3 reports the gain in final Class-IL accuracy over the unanchored objective for each β and K.

On the five-task sequence, compact frames support the widest range of anchoring strengths. With $K = 2$, every tested β improves accuracy, including a gain of 1.1 percentage points at $\beta = 0.80$. As the frame grows, weaker anchoring becomes more favorable: at the same mixing weight, $K = 80$ gives a change of -3.5 points. This ordering matches Theorem 1(ii): its sufficient condition $\beta < 1 - \sqrt{m/P}$ tightens as K grows and holds for no $\beta > 0$ once $K \geq 33$, where exact prototype matching is no longer excluded as a maximizer.

On the 20-task sequence, gains reach 2.5 and 2.4 points at $K = 5, \beta = 0.20$ and $K = 10, \beta = 0.50$, and on the 50-task sequence every configuration improves on the unanchored objective, by up to 4.0 points at $K = 5, \beta = 0.80$, a single-seed cell.

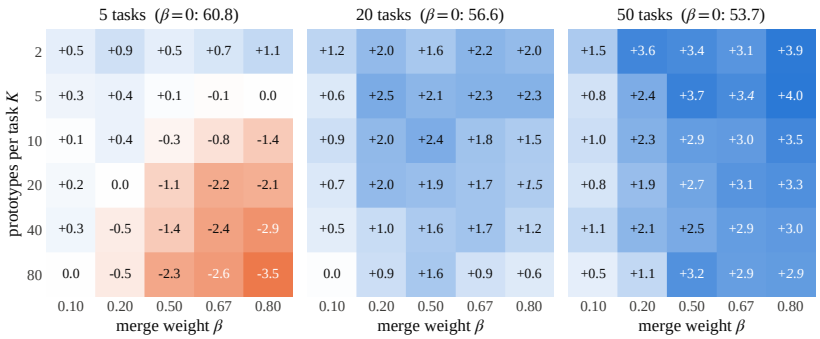

Figure 3: **Effect of the merge weight and the frame size.** Gain in final Class-IL accuracy over $\beta = 0$ (percentage points) for every merge weight β and number of prototypes per task K, at 5, 20 and 50 tasks. Mean over up to three seeds; italic values were not run and are the mean of their neighbouring cells.

These results support compact frames as references across the tested task partitions, including settings in which the number of prototypes is smaller than the number of classes per task. The anchors of one frame span at most $K - 1$ directions, with a rank set by prototype occupancy (Lemma 1), so a compact frame constrains few directions. For fixed embeddings, every $\beta \in (0, 1)$ lowers the batch objective when $\|A\|_* < \|C\|_*$ (Proposition 2), so any benefit of anchoring arises through retention. We therefore select β and K jointly for each task partition.

7 Discussion

MMCR produces dispersed embeddings but leaves their orientation unconstrained. GYRO reduces this freedom by anchoring centroids to persistent, task-specific frames within a single nuclear norm and without labels. The effect on accuracy depends on the length of the task sequence: LUMP is ahead on the five-task CIFAR-100 sequence, while GYRO reaches the highest final accuracy among the compared methods on the longer sequences of both benchmarks. On Tiny ImageNet, replay alone falls below fine-tuning under the evaluated configuration, whereas GYRO and LUMP, which draw on the same reservoir, improve over it. This suggests that how replayed images enter the objective matters, not only whether a buffer is available.

Geometric mechanism. GYRO changes the relative geometry of previously seen images less than the other methods we measured, on both benchmarks. The theory offers one account: the merge adds a single term to the MMCR deficit, the anchor distortion, which grows as a replayed image moves away from its task's frame (Theorem 1(i)), and when the frame is small relative to the batch, exact prototype matching is not a global maximizer, so the prototypes act as references rather than targets (Theorem 1(ii), Lemma 1). Consistent with this, compact frames perform well, including frames with fewer prototypes than classes per task. Preserving geometry does not by itself guarantee accuracy, however. On Tiny ImageNet, fine-tuning retains earlier geometry better than ER or LUMP yet ends with lower accuracy, and LUMP comes close to GYRO in accuracy while changing the established manifolds the most.

Limitations and outlook. GYRO depends on replay: earlier frames enter the objective only through stored images, and without a buffer anchoring does not improve accuracy (Appendix B). With a single task and no replay, it lowers accuracy by 1.7 points, in line with Proposition 2. The preferred β and K vary with the number of tasks, some large-β cells have a single seed, and GYRO has not been run at the larger replay budgets of Table S7. Hyperparameters were selected on test accuracy for every method, ours included. The geometric measurements use one run per method, and the theory describes the objective on a fixed batch rather than the training dynamics. Two extensions seem natural: setting the merge weight from the data, for instance from how far a task's representation has moved, and applying persistent frames to other rotation-invariant objectives.

REFERENCES

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision*, 2018.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, 2020.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *IEEE/CVF International Conference on Computer Vision*, 2021.

Sungmin Cha and Taesup Moon. Sy-con: Symmetric contrastive loss for continual self-supervised representation learning. *arXiv preprint arXiv:2306.05101*, 2023.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2019a.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. Classification and geometry of general perceptual manifolds. *Physical Review X*, 8(3):031003, 2018.

Giacomo Cignoni, Andrea Cossu, Alexandra Gomez-Villa, Joost van de Weijer, and Antonio Carta. Cla: Latent alignment for online continual self-supervised learning. In *Conference on Lifelong Learning Agents*, 2025.

Uri Cohen, SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature Communications*, 11:746, 2020.

Trung-Anh Dang, Vincent Nguyen, Ngoc-Son Vu, and Christel Vrain. Memory-efficient continual learning with neural collapse contrastive. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025.

Enrico Fini, Victor G. Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999.

Alex Gomez-Villa, Bartlomiej Twardowski, Lu Yu, Andrew D. Bagdanov, and Joost van de Weijer. Continually learning self-supervised representations with projected functional regularization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022.

Alex Gomez-Villa, Bartlomiej Twardowski, Kai Wang, and Joost van de Weijer. Plasticity-optimized complementary networks for unsupervised continual learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, 2016.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017.

Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations*, 2022.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Francesco Mezzadri. How to generate random matrices from the classical compact groups. *Notices of the American Mathematical Society*, 54(5):592–604, 2007.

Nikhil Shivakumar Nayak, Krishnateja Killamsetty, Ligong Han, Abhishek Bhandwaldar, Prateek Chanda, Kai Xu, Hao Wang, Aldo Pareja, Oleg Silkin, Mustafa Eyceoz, and Akash Srivastava. Sculpting subspaces: Constrained full fine-tuning in llms for continual learning. *arXiv preprint arXiv:2504.07097*, 2025.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021.

Rylan Schaeffer, Victor Lecomte, Dhruv Bhandarkar Pai, Andres Carranza, Berivan Isik, Alyssa Unell, Mikail Khona, Thomas Yerxa, Yann LeCun, SueYeon Chung, Andrey Gromov, Ravid Shwartz-Ziv, and Sanmi Koyejo. Towards an improved understanding and utilization of maximum manifold capacity representations. *arXiv preprint arXiv:2406.09366*, 2024.

Minhyuk Seo, Hyunseo Koh, Wonje Jeung, Minjae Lee, San Kim, Hankook Lee, Sungjun Cho, Sungik Choi, Hyunwoo Kim, and Jonghyun Choi. Learning equi-angular representations for online continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Joan Serrà, Dídac Surís, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, 2018.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Zheng Wang, Wanhao Yu, Li Yang, and Sen Lin. Rethinking continual learning with progressive neural collapse. In *International Conference on Learning Representations*, 2026.

Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *International Conference on Learning Representations*, 2023.

Thomas Yerxa, Yilun Kuang, Eero Simoncelli, and SueYeon Chung. Learning efficient coding of natural images with maximum manifold capacity representations. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, 2021.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 2017.

Yipeng Zhang, Laurent Charlin, Richard Zemel, and Mengye Ren. Integrating present and past in unsupervised continual learning. In *Conference on Lifelong Learning Agents*, 2024.

Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Appendix

Table of Contents

A GYRO: algorithm and derivations **2**
- A.1 Frame construction 2
- A.2 Proof of Theorem 1 2
- A.3 Frame-preserving transformations and the choice of β 3
- A.4 Supplementary statements 4

B Design ablations **5**

C Hyperparameters **6**

D Baseline Loss Formulations **6**

E Additional Results **9**

F Supplementary Results **9**
- F.1 Measuring Disruption of Learned Geometry 9

G Hyperparameter sweeps **11**

H Hyperparameter sweeps on Split CIFAR-100 **15**

A GYRO: ALGORITHM AND DERIVATIONS

Algorithm 1: GYRO (one run)

Input: tasks $\mathcal{D}_1, \ldots, \mathcal{D}_T$; frame size K; merge weight β; views V; batch sizes $n_{\text{cur}}, n_{\text{rep}}$; reservoir size $|\mathcal{M}|$

$\mathcal{M} \leftarrow \emptyset$;
for $t \leftarrow 1$ **to** T **do**
  if $t > 1$ **then** draw E_t by Equation (9); freeze it;
  for *each epoch* e *of task* t **do**
    if $t = 1$ ***and*** $e = 2$ **then** draw E_1; freeze it;
    for *each step* **do**
      sample n_{cur} images of $\mathcal{D}_t$, set $s_i \leftarrow t$;
      if $\mathcal{M} \neq \emptyset$ **then** add n_{rep} images of $\mathcal{M}$ with their stored s_i;
      compute the V views and the centroids c_i, Equation (1);
      if E_{s_i} *exists* **then**
        $a_i \leftarrow \operatorname{sg}\big(\arg\max_{e \in E_{s_i}} \langle c_i, e \rangle\big)$, ties to the lowest index;
        $y_i \leftarrow (1-\beta)c_i + \beta a_i$;
      else
        $y_i \leftarrow c_i$; `// first epoch of task 1 only`
      take an optimizer step on $-\|Y\|_*$;
  update $\mathcal{M}$ by reservoir sampling over $\mathcal{D}_t$, storing $s_i = t$ with each image;

A.1 FRAME CONSTRUCTION

Let $U \in \mathbb{R}^{K \times (K-1)}$ have orthonormal columns spanning $\mathbf{1}^{\perp}$, and let $Q_t \in \mathbb{R}^{D \times (K-1)}$ have orthonormal columns. The frame of task t is

$$E_t = \sqrt{\tfrac{K}{K-1}}\, U Q_t^{\top}. \tag{9}$$

Then $E_t E_t^{\top} = \frac{K}{K-1} U U^{\top} = \frac{K}{K-1}\big(I_K - \frac{1}{K}\mathbf{1}\mathbf{1}^{\top}\big)$, which gives unit rows, pairwise inner products $-\frac{1}{K-1}$ and $\mathbf{1}^{\top} E_t = 0$. The row span of E_t is the column span of Q_t, of dimension $K-1$, and the frame is tight on it: E_t has $K-1$ equal nonzero singular values $\sqrt{K/(K-1)}$, so $\|E_t\|_* = \sqrt{K(K-1)}$. When $K - 1 < D$, this span is a proper subspace of $\mathbb{R}^D$. For K unit rows the bound of Section 4.3 is $\sqrt{K \min(K, D)}$, so a simplex reaches a fraction $\sqrt{1 - 1/K}$ of it for $K \leq D$ (95% at $K = 10$) and the bound itself at $K = D + 1$.

Sampling. Our implementation draws a $D \times K$ Gaussian matrix G_t with a per-task seed, takes the orthonormal factor $\tilde{Q}_t$ of its QR decomposition, and sets $E_t = \sqrt{\frac{K}{K-1}}\big(I_K - \frac{1}{K}\mathbf{1}\mathbf{1}^{\top}\big)\tilde{Q}_t^{\top}$, which is Equation (9) with $Q_t = \tilde{Q}_t U$. Frames of different tasks are therefore random and independent. Fixing the signs of the QR factor so that its triangular factor has a positive diagonal makes $\tilde{Q}_t$ exactly uniformly distributed over matrices with orthonormal columns (Mezzadri, 2007); the simplex properties above hold for every orientation, and the dimension count in the proof of Proposition 1 holds with probability one for Gaussian sampling.

A.2 PROOF OF THEOREM 1

Throughout, the anchors depend on C only through $\operatorname{sg}(C)$ and on the frames of the tasks in the batch.

(i) Decomposition. Give the V views of image i weight $\frac{1-\beta}{V}$ each and its anchor weight β. The weights sum to one, their weighted mean is y_i, and every point has unit norm, so the weighted spread around y_i is $\sum_x w_x \|x - y_i\|_2^2 = \sum_x w_x \|x\|_2^2 - \|y_i\|_2^2 = 1 - \|y_i\|_2^2$. Split the points into two groups, the views (total weight $1 - \beta$, mean c_i, spread v_i by Equation (2)) and the anchor (weight β, spread 0). By the law of total variance the spread is the within-group part $(1-\beta)v_i$ plus the between-group

part $(1-\beta)\|c_i - y_i\|_2^2 + \beta\|a_i - y_i\|_2^2$. Since $c_i - y_i = \beta(c_i - a_i)$ and $a_i - y_i = (1-\beta)(a_i - c_i)$, the between-group part is $\beta(1-\beta)\|c_i - a_i\|_2^2$, and

$$1 - \|y_i\|_2^2 = (1-\beta)\big(1 - \|c_i\|_2^2\big) + \beta(1-\beta)\|c_i - a_i\|_2^2. \tag{10}$$

As $\|a_i\|_2 = 1$, $\|c_i - a_i\|_2^2 = 1 + \|c_i\|_2^2 - 2\nu_i$. Cauchy–Schwarz gives $\nu_i \le \|c_i\|_2$, with equality when c_i is a nonnegative multiple of a prototype. The rows of a simplex sum to zero (Section 4.1), so $\sum_k \langle c_i, e_{s_i,k}\rangle = 0$ and the largest term satisfies $\nu_i \ge 0$. If $\nu_i = 0$, every term is at most 0 and they sum to 0, so all vanish and $c_i \perp \operatorname{span} E_{s_i}$; the converse is immediate. Consequently $(1 - \|c_i\|_2)^2 \le \|c_i - a_i\|_2^2 \le 1 + \|c_i\|_2^2$.

Averaging Equation (10) over the P rows gives $1 - \|Y\|_F^2/P = q_{\text{view}} + q_{\text{anchor}}$. The definition $\Delta(Y) = r\|Y\|_F^2 - \|Y\|_*^2$ gives $\|Y\|_F^2/P = \|Y\|_*^2/(Pr) + q_{\text{spec}}$. Substituting yields Equation (8). The identity $\Delta(Y) = \sum_{j<\ell}(\sigma_j - \sigma_\ell)^2$ follows from expanding $r\sum_j \sigma_j^2 - (\sum_j \sigma_j)^2$ over the r singular values. Rows that have no anchor yet, which occurs only in the first epoch of task 1, are the $\beta_i = 0$ case of Equation (10); the identity then holds with a per-row weight β_i in place of β. Part (i) holds for any tie-breaking rule. □

(ii) Exact prototype matching. If $c_i = a_i$ for every row, then a_i is its own nearest prototype, since $\langle a_i, a_i\rangle = 1 > \langle a_i, e\rangle$ for every other row e of the frame, and $y_i = (1-\beta)a_i + \beta a_i = a_i$, so $Y = A$. Cauchy–Schwarz over the at most m nonzero singular values gives $\|A\|_*^2 \le m\|A\|_F^2 = mP$. For the second claim, the nuclear norm is dual to the operator norm, so $\|Y\|_* \ge \langle Y, W\rangle$ for every W with $\|W\|_{\text{op}} \le 1$. If C has orthonormal rows, which requires $P \le D$, then $\|C\|_{\text{op}} = 1$ and

$$\|Y\|_* \ge \langle Y, C\rangle = (1-\beta)\|C\|_F^2 + \beta\sum_i \langle a_i, c_i\rangle = (1-\beta)P + \beta\sum_i \nu_i \;\ge\; (1-\beta)P,$$

using $\nu_i \ge 0$ from part (i). Exact matching therefore gives a strictly smaller value than any such C when $(1-\beta)P > \sqrt{Pm}$, that is, $(1-\beta)^2 > m/P$. For $\operatorname{rank} A \le m$: the current anchors are rows of E_t, whose span has dimension $K-1$, each of the n_{rep} replayed rows raises the rank by at most one, and $\operatorname{rank} A \le \min(P, D)$ for any $P \times D$ matrix. □

A.3 Frame-preserving transformations and the choice of β

Rows are row vectors, so an orthogonal map acts as $x \mapsto xR$ and $\langle xR, yR\rangle = \langle x, y\rangle$.

Proposition 1 (Frame-preserving transformations)**.** *Fix a batch and $\beta \in [0, 1)$. The loss depends on the frames only through $\{E_s : s \in \mathcal{T}_B\}$. Let $R \in O(D)$ map the row set of each of these frames onto itself. If the assignment in Equation (4) has no ties, then $\mathcal{L}_{\text{GYRO}}(CR) = \mathcal{L}_{\text{GYRO}}(C)$, with the anchors at CR recomputed from the same frames. Without replay $\mathcal{T}_B = \{t\}$, and the group of such R is isomorphic to $S_K \times O(D-K+1)$, where S_K permutes the K prototypes. Averaged over reservoir draws, the expected loss is unchanged by every R that preserves E_t and the frame of each task with images in the reservoir.*

Proof. Equation (4) searches only E_{s_i} for row i, and $s_i \in \mathcal{T}_B$, so no other frame enters the loss. Take R that maps the row set of each E_s, $s \in \mathcal{T}_B$, onto itself. For an image with frame E_s and centroid c, $\langle cR, eR\rangle = \langle c, e\rangle$ for every row e, so the row of E_s that maximizes the inner product with cR is aR, where a maximizes it with c; aR is a row of E_s by the choice of R. With no ties the assignment is unique, so the anchor of cR is aR and $(1-\beta)cR + \beta aR = yR$. Stacking the rows gives $Y(CR) = Y(C)R$, and $\|Y(C)R\|_* = \|Y(C)\|_*$.

Without replay the group consists of the R that map the rows of E_t onto themselves. Such an R maps $W = \operatorname{span} E_t$, of dimension $K-1$, onto itself and $W^\perp$ onto itself, so $R = R_W \oplus R_\perp$. On W, R_W is determined by the permutation it induces on the K rows, which span W; conversely every permutation of the vertices of a regular simplex preserves all inner products of the frame (Section 4.1) and so extends to an isometry of W. The restrictions therefore form S_K. On $W^\perp$ no row constrains $R_\perp$. Hence the group is isomorphic to $S_K \times O(D-K+1)$ for every t, while its orientation follows E_t.

Expected loss. Let $S_t^{\mathcal{M}}$ be the span of E_t and of the frames of every task with images in the reservoir, and $G_t^{\mathcal{M}}$ the maps that send each of these frames onto itself. Every batch that can be drawn uses only such frames, so by the argument above the expected loss over reservoir draws is invariant under $G_t^{\mathcal{M}}$.

This is a sufficient condition, and other maps may also leave the expected loss unchanged. The same argument as for one frame gives $G_t^{\mathcal{M}} \cong F_t \times O(D - \dim S_t^{\mathcal{M}})$ with F_t finite: an element is fixed on $S_t^{\mathcal{M}}$ by where it sends a spanning set of rows, of which there are finitely many choices. The row span of E_s is the column span of Q_s, of dimension $K-1$, and the frames are functions of independent Gaussian matrices (Appendix A.1), so the sum of their spans has dimension equal to the smaller of D and the sum of their dimensions with probability one. So $\dim S_t^{\mathcal{M}} = \min\{D,\, n_t(K-1)\}$ almost surely, where n_t is the number of frames involved. With a 2000-image reservoir every earlier task keeps images in all our runs, so $n_t = t$ and $G_{t+1}^{\mathcal{M}} \subseteq G_t^{\mathcal{M}}$; a task that leaves the reservoir would remove its constraint. Rotations that act only on the orthogonal complement of the joint span of the frames keep every prototype in place; when this complement has dimension at least two, they form a continuous family. □

Proposition 2 (Endpoint maximization in β)**.** *For fixed C and anchors, $\beta \mapsto \|Y\|_*$, extended to $[0,1]$, is convex, so it attains its maximum at an endpoint. If $\|A\|_* < \|C\|_*$, then $\|Y\|_* < \|C\|_*$ for every $\beta \in (0,1)$.*

Proof. The anchors do not depend on β, so $Y(\beta) = (1-\beta)C + \beta A$ is affine in β and $\beta \mapsto \|Y(\beta)\|_*$ is convex as a norm of an affine map; a convex function on $[0,1]$ attains its maximum at an endpoint. By the triangle inequality, $\|Y(\beta)\|_* \le (1-\beta)\|C\|_* + \beta\|A\|_*$, which is strictly below $\|C\|_*$ for $\beta \in (0,1)$ when $\|A\|_* < \|C\|_*$. Convexity leaves room for maximizers in the interior when the function is constant, for example at $C = A$; the value $\beta = 1$ enters only as an endpoint of this analysis. □

A.4 Supplementary statements

Anchor rank and prototype occupancy. The rank bound m of Theorem 1 holds for every assignment. The rank that the anchors realize depends on how the images of a batch occupy the prototypes, and for a single frame it has an explicit description.

Lemma 1 (Anchor rank of one frame)**.** *Let the P rows of A be prototypes of one frame E, with n_j rows on prototype j, $p_j = n_j/P$, $p_{\max} = \max_j p_j$, k occupied prototypes, and $H = \big(\sum_j \sqrt{p_j}\big)^2$ the Hill number of order $\frac{1}{2}$ of the occupancy. The effective anchor rank $m_A = \|A\|_*^2/P$ satisfies*

$$\max\Big\{1,\ \tfrac{K}{K-1}\big(\sqrt{H} - \sqrt{p_{\max}}\big)^2\Big\} \ \le\ m_A \ \le\ \min\Big\{\tfrac{K}{K-1}H,\ K-1,\ k\Big\}.$$

At uniform occupancy, $p_j = 1/K$, both bounds equal $K-1$, so $m_A = K-1$. For $K \ge 3$, $m_A = K-1$ holds only at uniform occupancy.

Proof. Write $A = SE$ with $S \in \{0,1\}^{P\times K}$ the assignment and $N = S^\top S = \mathrm{diag}(n)$. The nonzero eigenvalues of $AA^\top = SEE^\top S^\top$ are those of $N^{1/2}EE^\top N^{1/2}$, and $EE^\top = \frac{K}{K-1}\big(I_K - \frac{1}{K}\mathbf{1}\mathbf{1}^\top\big)$ (Appendix A.1), so $N^{1/2}EE^\top N^{1/2} = \frac{K}{K-1}\big(N - \frac{1}{K}uu^\top\big)$ with $u_j = \sqrt{n_j}$. Let $\lambda_1 \ge \cdots \ge \lambda_K \ge 0$ be the eigenvalues of $N - \frac{1}{K}uu^\top$; then $\|A\|_* = \sqrt{K/(K-1)}\,\sum_k \sqrt{\lambda_k}$.

Upper bounds. $N - \frac{1}{K}uu^\top \preceq N$ and the matrix square root is operator monotone, so $\sum_k \sqrt{\lambda_k} \le \mathrm{tr}\, N^{1/2} = \sum_j \sqrt{n_j}$, which gives $m_A \le \frac{K}{K-1}H$. Cauchy–Schwarz over the nonzero singular values gives $\|A\|_*^2 \le \mathrm{rank}(A)\,\|A\|_F^2 = \mathrm{rank}(A)\,P$, and $\mathrm{rank}\,A$ is at most $K-1$, the dimension of the span of E, and at most k.

Lower bounds. $\|A\|_* \ge \|A\|_F = \sqrt{P}$ gives $m_A \ge 1$. Subtracting a rank-one positive semidefinite matrix interlaces the eigenvalues: with $n_{(1)} \ge \cdots \ge n_{(K)}$ the sorted counts, $\lambda_k \ge n_{(k+1)}$ for $k < K$. Hence $\sum_k \sqrt{\lambda_k} \ge \sum_j \sqrt{n_j} - \sqrt{n_{(1)}}$, and squaring gives the second lower bound. At uniform occupancy it equals $\frac{K}{K-1}\big(\sqrt{K} - 1/\sqrt{K}\big)^2 = K-1$, which matches the upper bound.

Equality. $m_A = K-1$ requires equality in Cauchy–Schwarz with $\mathrm{rank}\,A = K-1$: all K prototypes occupied and a flat nonzero spectrum, so $\frac{K}{K-1}\big(N - \frac{1}{K}uu^\top\big)$ is a multiple of a rank-$(K-1)$ projector. Comparing off-diagonal entries forces $n_j n_\ell$ to be constant over all pairs $j \ne \ell$, which for $K \ge 3$ means uniform occupancy. □

In $\sqrt{m_A}$, the two Hill-number bounds differ by at most $\sqrt{Kp_{\max}/(K-1)}$, so when occupancy is spread, m_A is close to $\frac{K}{K-1}H$: the anchor rank follows the Hill number of the occupancy and reaches

$K-1$ at uniform occupancy. When one prototype holds most rows, the bounds $1 \le m_A \le k$ are the informative ones. With replay, rows anchored to other frames can add up to n_{rep} further directions, as in the bound m.

The neural-collapse regime. If $K = D + 1$, $P \ge K$ and occupancy is uniform, then $\operatorname{rank} A = D = r$ and $\|A\|_*^2 = (K-1)P = rP$, so the collapsed batch attains the MMCR bound $\sqrt{Pr}$ of Section 4.3. In this configuration the frame is an optimum of the objective, one of possibly several, rather than a reference. When $K - 1 < r$, as in all our runs, Theorem 1(ii) applies instead. This compares geometric configurations and makes no statement about the dynamics of neural-collapse methods.

Thresholds for our batch composition. Theorem 1(ii) applies when $\beta < 1 - \sqrt{m/P}$ with $m = \min\{P, D, K-1+n_{\text{rep}}\}$. For the CIFAR-100 batches with replay, $P = 40$, $n_{\text{rep}} = 8$ and $D = 128$, the threshold is approximately 0.526, 0.452, 0.348 and 0.178 at $K = 2$, 5, 10 and 20. On the first task, $P = 32$ and $n_{\text{rep}} = 0$, it is approximately 0.823, 0.646, 0.470 and 0.229. The configuration $\beta = 1/3$ therefore satisfies the sufficient condition for $K \le 10$, and $K = 20$ lies outside it.

Random-subspace intuition. The row span of E_s is the column span of Q_s, drawn independently for each task. For uniformly distributed frames, a unit vector chosen independently of a random $(K-1)$-dimensional subspace has expected squared projection $(K-1)/D$ onto it, so independent frames are close to orthogonal when $K-1 \ll D$. This describes the frames themselves; how closely learned representations follow their frames is an empirical question, examined in Section 6.

A sharper condition for a single frame. For a block of P rows anchored to one frame, with K dividing P and $D \ge P + K - 1$, collapse is not a maximizer for any $\beta \in [0, 1)$ as soon as $x = (K-1)/P < (3-\sqrt{5})/2 \approx 0.382$. Take orthonormal $u_i \perp \operatorname{span} E$, choose prototypes with uniform occupancy, and set $c_i = \sqrt{1-\varepsilon^2}\, u_i + \varepsilon a_i$; for $\varepsilon > 0$ the assignment returns a_i without ties, and as $\varepsilon \to 0$, $YY^\top \to (1-\beta)^2 I_P + \beta^2 AA^\top$. By Lemma 1 and its proof, the limit has singular values $1-\beta$ (multiplicity $P-K+1$) and $\sqrt{(1-\beta)^2 + \beta^2 P/(K-1)}$ (multiplicity $K-1$), so $f(\beta) = (P-K+1)(1-\beta) + (K-1)\sqrt{(1-\beta)^2 + \beta^2 P/(K-1)}$. The function f is convex with $f(1) = \sqrt{P(K-1)}$, the largest collapsed value, and $f'(1) = \sqrt{P(K-1)} - (P-K+1)$, which is negative exactly when $\sqrt{x} < 1 - x$. A convex function with negative slope at 1 is strictly decreasing on $[0, 1]$, so $f(\beta) > \sqrt{P(K-1)}$ for all $\beta < 1$. With replay the anchor spectrum is no longer flat, and the general condition of Theorem 1(ii) is the one that holds.

Effect of β on the update. For locally fixed assignments, the anchors are constants, so the gradient of $-\|Y\|_*$ with respect to C is $-(1-\beta)\, U_Y V_Y^\top$ for the compact singular value decomposition $Y = U_Y \Sigma_Y V_Y^\top$ of a full-rank Y; at rank deficiency the nuclear norm has a set of subgradients. The parameter gradient then passes through the centroid average, the normalization and the network. Multiplying every gradient by a constant $c > 0$ multiplies Adam's first-moment estimate by c and its second by c^2, which leaves the update unchanged up to the ϵ in its denominator, the weight decay (10^{-6}) added to the gradient, and an optimizer history that contains differently scaled gradients, such as those of the unanchored first epoch. Changing β also changes the direction $U_Y V_Y^\top$, since the anchors tilt the singular vectors of Y. Under Adam, the effect of β therefore extends beyond a change of step size.

B Design ablations

Table S1 isolates the design choices of Section 4.2 in a smaller setting: Split CIFAR-100 with a ResNet-18 backbone, $K = 10$, 20 epochs per task, and the replay budget of the main experiments. Values are final Class-IL accuracy, mean±std over seeds.

These runs support the choices in the main text, within the tested settings. Consistent with the locality of the objective (Proposition 1), the gains from anchoring appear together with replay: without a buffer, anchoring changes accuracy by -0.3 to -0.9 points, and with one it adds 3.5 points at 20 tasks and 5.2 at 50. Current images account for 80% of the batch rows (32 of 40), and anchoring them establishes alignment with the frame later used during replay; once replayed images are

Table S1: GYRO design ablations, Split CIFAR-100, ResNet-18, $K = 10$. "Anchor current" and "anchor replayed" give the merge weight on each block. The shared frame uses one frozen K-row frame for every task instead of a new frame per task.

Variant	5 tasks	10 tasks	20 tasks	50 tasks
MMCR, no replay	54.2±0.3	50.9±0.6	46.2±0.7	39.0±0.2
Anchoring, no replay ($\beta = 1/3$)	53.3±0.5	50.4±0.3	45.4±0.6	38.7±0.2
Replay, no anchoring ($\beta = 0$)	55.9±0.5	53.4±0.3	50.0±0.2	45.5±0.5
Anchor replayed only ($1/3$), current 0	—	—	52.1±0.6	48.7±1.5
Shared frame for all tasks	55.2	52.5	49.7	44.9
Constant $\beta = 0.6$	—	—	53.3±0.3	51.6±0.6
Replayed 0.6, current ramped to 0.2	—	—	53.4±0.5	49.6±0.4
GYRO ($\beta = 1/3$ on both blocks)	55.5±0.3	55.0±0.6	53.5±0.5	50.7±0.9

anchored, anchoring current images adds 1.4 and 2.0 points at 20 and 50 tasks. In the reported ablation, sharing one frame across tasks did not improve on replay alone, which supports anchoring each replayed image to the frame of its own task. The tested separate-weight and ramp schedules did not consistently improve on a shared constant weight. With a single task and no replay, anchoring lowers accuracy by 1.7 points (56.7 against 58.5), in line with the trade-off between current-task adaptation and retention discussed in Section 4.3. Rescaling the learning rate by $1/(1-\beta)$ changes accuracy by at most 0.5 points, consistent with β acting through more than the effective step size.

C Hyperparameters

This section collects every hyperparameter used in the paper. Table S2 covers the natural-image benchmarks, Table S3 the method-specific settings, and Table S4 the synthetic benchmark.

The CIFAR-100 budget of 20 epochs per task matches the epoch sweep of Section F.1.1, which locates the best operating point at 10–20 epochs per task, and is the budget used for Table 1.

D Baseline Loss Formulations

Every method shares the same backbone, projector and MMCR objective, and differs only in what it adds. Write $z_{i,v} \in \mathbb{S}^{D-1}$ for the embedding of view v of image i, $c_i = \frac{1}{V}\sum_{v=1}^{V} z_{i,v}$ for its centroid, and $C \in \mathbb{R}^{P\times D}$ for the matrix stacking the $c_i^\top$ of a batch. The shared objective is the negative nuclear norm

$$\mathcal{L}_{\mathrm{MMCR}} = -\|C\|_* = -\sum_j \sigma_j(C), \tag{11}$$

and each method below contributes an additional term, a modification of the rows entering equation 11, or a projection applied to the gradient. Terms are inactive on the first task, where no previous task exists to anchor to. Every term computed in embedding space, including GYRO's anchoring, acts on the projector output; EWC acts on the parameters and the projection methods on the gradients. The reservoir stores raw images, which the current network re-encodes at every replay, and replayed images pass through the encoder in a separate forward pass with their own BatchNorm statistics, as for every replay baseline. Evaluation reads the backbone feature, from a linear probe trained by a separate Adam optimizer.

Fine-Tune Equation (11) alone, with no continual mechanism.

ER (Naive + Replay). No additional term. Replayed images are embedded in a separate forward pass, so batch normalization uses their own batch statistics, and their centroids are appended as extra rows of the centroid matrix: $\mathcal{L} = -\left\| [\, C_{\mathrm{live}} \,;\, C_{\mathrm{replay}} \,] \right\|_*$.

Table S2: Benchmark, architecture, optimization, and linear-probe settings, per dataset. A value spanning both columns is shared by the two benchmarks.

Hyperparameter	**Split CIFAR-100**	**Split Tiny ImageNet**	**Source**
Benchmark			
Classes	100	200	§5.1
Train / test images	50,000 / 10,000	100,000 / 10,000	§5.1
Image size	32×32 RGB	64×64 RGB	§5.1
Task splits	5×20, 10×10, 20×5, 50×2	20×10, 50×4, 100×2 (200×1 in App.)	§5.1
Train augmentations	random-resized-crop, color jitter, random grayscale, horizontal flip		§5.1
Eval transform	none (full normalized image, no crop)		§5.1
Augmentation (independently per view)			
Random resized crop	32 px, scale 0.08–1.0, aspect 3/4–4/3	64 px, scale 0.08–1.0, aspect 3/4–4/3	§5.1
Horizontal flip	$p = 0.5$		§5.1
Color jitter	brightness, contrast, saturation 0.4; hue 0.1; applied with $p = 0.8$		§5.1
Random grayscale	$p = 0.2$		§5.1
Blur / solarization	not used		§5.1
Architecture			
Backbone	ResNet-50, trained from scratch		§5.1
Stem	3×3 stride-1, no max-pool	7×7 stride-2, 3×3 max-pool	§5.1
Backbone feature dim.	2048		§5.1
Embedding dimension D	128	256	§5.1
Projector hidden dim.	512		§5.1
Projector shape	2048→512→128	2048→512→256	§5.1
Loss / eval readout	projector output / backbone feature		§5.1
Batch normalization	enabled		§5.1
Embedding normalization	ℓ_2, unit sphere		§5.1
Optimization			
Optimizer	Adam		§5.1
Learning rate	10^{-3}, constant, no warm-up		§5.1
Weight decay	10^{-6}		§5.1
Gradient clipping	global norm 1.0		§5.1
Task-boundary state	optimizer and BatchNorm state carried over		§5.1
Batch size	32		§5.1
Views per image V	40		§5.1
Epochs per task	20	20	§5.1
Seeds	{78974, 45466, 3345}		§5.1
Linear probe (evaluation only)			
Optimizer	Adam (separate from encoder)		§5.1
Learning rate	10^{-2}		§5.1
Batch size	512		§5.1
Epochs	50, cosine annealing		§5.1
Weight decay	10^{-6}		§5.1
Readout feature	frozen 2048-d backbone feature		§5.1
Fit protocol	refit on frozen encoder at each task boundary		Table S8

Table S3: Method-specific hyperparameters. GYRO adds only K and β to the shared configuration of Table S2. Per-task-count values are for 5 / 20 / 50 tasks on CIFAR-100 and 20 / 50 / 100 tasks on Tiny ImageNet; the Tiny ImageNet search is in Section G. A value spanning both columns is dataset-independent.

Hyperparameter	**Split CIFAR-100**	**Split Tiny ImageNet**	**Source**
GYRO			
Prototypes per task K	2 / 20 / 5	5 / 10 / 20	§4
default (untuned)	10		§4
Mixture coefficient β	$1/3$	$1/3$ / $1/3$ / 0.5	§4
equivalent α	0.5, $\beta = \frac{\alpha}{1+\alpha}$	0.5 / 0.5 / 1	§4
Frame type	per-task simplex ETF, fixed by (K, D) and a per-task seed		§4.1
Frame storage	$O(TKD)$		§4.1
Assignment	nearest prototype, recomputed every step		§4
Assignment cost	$O(PKD)$ per batch, no extra encoder pass		§4
Replay (where used)			
Reservoir size	2000 images		§4
Current images per step	32		§4
Replay images per step	8		§4
First task	current-image batch only		§4
Replay fraction (baselines)	25% of accumulated old-task samples		*Analysis*
Baseline settings selected by sweep (per task count)			
EWC λ	0.3 / 0.03 / 0.03	10 / 10 / 0.03	§D
CaSSLe α	0.03 / 0.01 / 100	1 / 1 / 1	§D
DER α	3.0 / 3.0 / 10.0	1 / 1 / 3	§D
DER++ α	—	1 / 1 / 1	§D
NGP rank	1 / 8 / 2	8 / 8 / 8	§D
ParamProj preserve-frac.	0.01 / 0.01 / 0.01	0.01 / 0.1 / 0.1	§D
LUMP a (8 replay/step)	0.4 / 1.0 / 0.4	1 / 1 / 1	§D
LUMP a (32 replay/step)	2.0 / 8.0 / 16.0	not run	§D
DER++ replay/step	16	8 (shared reservoir)	§D

Table S4: Synthetic benchmark. The two embedding dimensions correspond to the double-descent bottleneck ($D = 8$) and the properly-scaled configuration ($D = 32$); batch size is fixed at $8D$ in both.

Hyperparameter	**Value**	**Source**
Classes	10	*Analysis*
Input dimension	20	*Analysis*
Task split	5 tasks $\times$ 2 classes	*Analysis*
Shared-noise level σ	1.5	*Analysis*
Backbone	6-hidden-layer MLP	*Analysis*
Views per sample n_{aug}	16	*Analysis*
Embedding dimension D	8 (bottleneck) / 32 (main)	*Analysis*
Batch size	$8D$: 64 (D=8) / 256 (D=32)	*Analysis*
Seeds	{42, 7, 123}	*Analysis*
GPM directions per task	70	*Analysis*

EWC. A quadratic penalty on parameter drift, with the diagonal Fisher F_θ estimated at each task boundary and accumulated across tasks,

$$\mathcal{L} = \mathcal{L}_{\mathrm{MMCR}} + \lambda_{\mathrm{EWC}} \sum_{\theta} F_\theta \, (\theta - \theta^\star)^2, \tag{12}$$

where $\theta^\star$ are the parameters at the last boundary. The implementation stores the expanded form, so the accumulated penalty is evaluated without retaining one anchor per task.

CaSSLe. A trainable predictor g (Linear–BatchNorm–ReLU–Linear) maps the live centroid onto the frozen one, and the penalty is a negative cosine,

$$\mathcal{L} = \mathcal{L}_{\mathrm{MMCR}} + \alpha \, \frac{1}{P} \sum_{i=1}^{P} \Big(1 - \big\langle \hat{g}(c_i),\, c_i^{\mathrm{old}} \big\rangle\Big), \qquad \hat{g}(c) = \frac{g(c)}{\|g(c)\|_2}. \tag{13}$$

Only the prediction is normalized; the target keeps its norm, so the penalty does not reach zero. The predictor separates CaSSLe from LwF's direct distillation.

DER. Each buffered exemplar stores the embedding it had when it was inserted. Those frozen vectors are regression targets for the current encoder,

$$\mathcal{L} = \mathcal{L}_{\mathrm{MMCR}} + \alpha_{\mathrm{DER}} \, \frac{1}{N_e D} \sum_{e=1}^{N_e} \big\| \hat{z}_e - t_e \big\|_2^2, \tag{14}$$

with t_e the stored target. Exemplars pass through the evaluation transform rather than the training augmentations, matching the transform under which t_e was recorded.

DER++. The DER penalty above, plus the shared replay of ER.

LUMP. No additional term. Each live view is interpolated with the matching view of a replayed image, using one $\lambda \sim \mathrm{Beta}(a, a)$ draw per step, and the objective is evaluated on the mixture alone:

$$\tilde{z}_{i,v} = \lambda \, z_{i,v}^{\mathrm{live}} + (1 - \lambda) \, z_{i,v}^{\mathrm{replay}}. \tag{15}$$

NGP and ParamProj. No additional term; each constrains the update instead. Writing G for a layer's gradient reshaped to [out] $\times$ [in] and s for a scale,

$$\text{NGP:} \quad G \leftarrow G - s \, Q^\top (QG), \tag{16}$$

$$\text{ParamProj:} \quad G \leftarrow G - (GV^\top)V. \tag{17}$$

GPM and ParamProj protect the row space of the stored bases, NGP the column space, that is, the output directions in which finished tasks' nuclear-norm gradients lay.

GYRO. No auxiliary term. Each centroid is mixed with its selected prototype before the objective is evaluated, so the fixed references enter the nuclear norm itself,

$$\mathcal{L}_{\mathrm{GYRO}} = -\big\|(1 - \beta)C + \beta A\big\|_*, \tag{18}$$

with A stacking the assigned anchors. At $\beta = 0$ this reduces to equation 11 exactly.

E Additional Results

F Supplementary Results

F.1 Measuring Disruption of Learned Geometry

Figure 2 measures how much a model rearranges what it has already learned. A set P of image pairs is drawn from previously seen tasks and held fixed. Let $\cos_{ij}^t$ be the cosine similarity between the backbone features of images i and j at task boundary t, the layer read by the linear probe. The

Table S5: Task-IL average and backward transfer for the runs of Table 1. **Avg** is the task-IL average; **BWT** is measured from each task's accuracy immediately after it was trained.

	5 tasks		20 tasks		50 tasks	
Method	**Avg↑**	**BWT↑**	**Avg↑**	**BWT↑**	**Avg↑**	**BWT↑**
Fine-tune	79.8±0.3	−1.9±0.1	88.4±1.0	−3.0±1.1	91.5±0.7‡	−3.1±0.3‡
EWC	80.0±0.1	−1.6±0.3	88.3±0.8	−2.9±0.6	90.6±0.9	−3.8±1.0
CaSSLe	79.9±0.2	−2.1±0.6	88.4±0.4	−3.1±0.3	93.2±1.4	**1.0±0.5**
NGP	80.1±0.4	−1.5±0.1	88.4±0.2	−3.4±0.0	91.3±0.6	−3.3±0.5
Param. Proj.	80.3±0.1	−1.7±0.4	89.1±0.1	−2.8±0.1	93.2±0.2	−2.1±0.3
ER	79.9±0.2	−1.4±0.5	89.6±0.3	−1.8±0.2	93.6±0.7	−1.2±0.4
DER	79.4±0.6	−1.2±1.0	88.5±0.3	−3.0±0.1	92.3±0.3	−2.7±0.2
DER++	79.6±0.5	**−0.4±0.5**	89.4±1.0	−2.1±0.7	93.4±0.6	−1.7±0.5
LUMP	**81.3±0.3**	**−0.4±0.3**	90.3±0.8	−2.0±0.5	93.9±0.8	−1.8±0.7
GYRO	80.2±0.2	−0.9±0.3	**91.2±0.2**	**−0.1±0.2**	**95.3±0.5**	0.0±0.3
Joint (UB)	—	—	—	—	—	—

Table S6: Task-IL average and backward transfer for the Split Tiny ImageNet runs of Table 2, each method at the same setting.

	20 tasks		50 tasks		100 tasks	
Method	**Avg↑**	**BWT↑**	**Avg↑**	**BWT↑**	**Avg↑**	**BWT↑**
Fine-tune	59.9±0.7	−9.8±0.4	72.9±1.0	−6.9±0.7	81.7±0.5	−6.0±0.4
EWC	60.3±0.6	−8.9±0.5	70.9±0.3	−7.8±0.3	81.5±0.6	−6.6±0.4
CaSSLe	60.5±0.7	−9.1±0.4	73.4±1.3	−6.2±1.0	81.5±0.2	−6.3±0.1
NGP	60.4±1.2	−9.5±1.1	73.5±0.2	−6.5±0.3	82.7±0.5	−5.3±0.3
Param. Proj.	60.1±0.6	−9.9±0.5	74.4±0.1	−6.0±0.2	85.9±0.2	−3.3±0.1
ER	55.3±0.5	−10.7±0.2	70.2±0.1	−7.3±0.0	83.1±0.1	−4.0±0.1
DER	61.0±0.5	−8.8±0.8	72.4±0.7	−7.9±0.6	82.7±0.5	−5.6±0.4
DER++	55.9±0.7	−10.1±0.6	70.5±0.0	−7.7±0.3	83.4±1.0	−3.7±0.9
LUMP	62.0±0.9	−8.3±0.7	74.4±0.8	−6.3±0.8	**86.3±0.6**	−3.1±0.7
GYRO	**63.1±0.4**	**−4.6±0.6**	**77.3±0.1**	**−1.5±0.1**	85.5±0.5	**−2.0±0.3**

Table S7: What a larger replay budget buys. Every row here replays 16 or 32 images per step against the 8 of Table 1, so these cells are not better-tuned versions of the matched rows: they see two to four times the data per step. Each should be read against that method's matched row rather than ranked against the others. Final Class-IL accuracy, mean±std over three seeds.

Method	**Setting**	**5 tasks**	**20 tasks**	**50 tasks**
LUMP	32 replayed/step	63.8±0.4	62.7±0.1	61.8±0.4
ER	16 replayed/step	62.0±0.4	57.5±0.1	55.7±0.1
DER++	16 replayed/step	61.3±0.4	57.0±0.2	55.4±0.2
GYRO	8 replayed/step	61.7±0.3	59.3±0.5	57.2±0.5

Table S8: **Continual-learning metrics reported in the main text and Appendix E.** $A_{i,j}$ is Task-IL accuracy on task j under the checkpoint saved after training task i, and T is the number of tasks.

Metric	Definition	Interpretation
ACC ↑	$\frac{1}{T}\sum_{j=1}^{T} A_{T,j}$	Mean accuracy over all tasks under the final model. The standard single-number headline metric.
Final ↑	$A_{T,T}$	Accuracy on the most recently learned task only. Contrasts with ACC, which averages over old tasks too; the gap between them is the cost of retention.
BWT ↑	$\frac{1}{T-1}\sum_{j=1}^{T-1}(A_{T,j} - A_{j,j})$	Change in accuracy on task j between learning it and finishing training. Negative means forgetting; equals $-$ `avg_forgetting` as logged by the training script.
F ↓	$\frac{1}{T-1}\sum_{j=1}^{T-1}(\max_{i<T} A_{i,j} - A_{T,j})$	Average forgetting: drop from the *best* accuracy task j ever reached, rather than from $A_{j,j}$. Identical to $-$BWT unless a task peaks after it was first learned, in which case $A_{j,j}$ understates what was lost.
K ↑	$\frac{1}{T-1}\sum_{i=2}^{T}(A_{i,i} - A_{i-1,i})$	Knowledge gain: improvement on task i from training on it, relative to the accuracy the model already had on task i just before it was introduced. Large K means each task is learned from scratch; small K means the representation already transferred.
Not computable from the stored results (see caption):		
FWT	$\frac{1}{T-1}\sum_{i=2}^{T}(A_{i-1,i} - R_i)$	Forward transfer: benefit of earlier tasks on task i relative to a random-init reference R_i.

disruption at boundary t is the mean squared change of these similarities across that boundary, and the figure reports its total over all boundaries,

$$D_T \;=\; \sum_{t\le T} \big\langle\, (\cos_{ij}^{t+1} - \cos_{ij}^{t})^2 \,\big\rangle_{(i,j)\in P}. \tag{19}$$

Solid bars restrict P to pairs from the same task and lighter hatched bars to pairs from different tasks. Because D_T compares cosines, it is invariant to any common rotation of the representation, which a refitted linear probe cannot observe either. D_T can only grow with T, and the gap between two methods at T is the disruption one has avoided relative to the other up to that point.

F.1.1 Fine-Tune epoch sweep and manifold geometry

Table S9: Naive MMCR epoch sweep, Split-CIFAR-100.

Backbone	Tasks	10 ep	20 ep	50 ep	100 ep
ResNet-50	5	58.7±0.4 / 77.9 / **0.6**	**60.6±0.2** / 79.5 / 2.2	58.4±0.5 / 78.5 / 5.0	54.2±0.8 / 76.2 / 7.0
ResNet-50	20	53.2±0.1 / 87.1 / **2.3**	**53.4±0.3** / 88.9 / 2.7	50.0±0.4 / 87.9 / 4.1	48.6±0.3 / 87.3 / 3.7
ResNet-18	5	51.5±0.1 / 73.4 / **1.4**	**52.8±0.4** / 74.6 / 3.6	49.6±0.2 / 72.1 / 8.7	—
ResNet-18	20	**45.9±0.4** / 85.9 / **2.6**	44.9±0.3 / 85.8 / 4.4	38.8±0.3 / 83.1 / 7.4	—

Final Class-IL peaks at 10–20 epochs/task and falls with more training, while forgetting rises nearly monotonically (ResNet-50, 5 tasks: $0.6 \to 2.2 \to 5.0 \to 7.0$). ResNet-50 at 20 epochs beats its own 100-epoch runs by $+6.4$ (5 tasks) and $+4.8$ (20 tasks) points of Final accuracy.

F.1.2 CIFAR-100, 20 epochs/task

100 classes, ResNet-50, embedding dimension 128, re-run at the epoch budget Section F.1.1 identifies as the better operating point.

G Hyperparameter sweeps

This subsection documents every hyperparameter search behind the Split Tiny ImageNet numbers of Table 2: which knob was searched for each method, over which values, how the value was chosen, and what each value scored.

Table S10: JAX baselines, Split-CIFAR-100, 5 tasks $\times$ 20 classes, 20 epochs/task. ‡ Fewer than 3 seeds.

Method	**Final** $\uparrow$	**Avg** $\uparrow$	**Forg** $\downarrow$
GPM	60.6±0.6	79.5±0.3	1.9±0.4
Naive (MMCR)	60.6±0.2	79.5±0.4	2.2±0.1
CaSSLe	60.2±0.4	79.3±0.4	2.3±0.6
LwF‡	60.2±0.7	79.5±0.5	1.2±0.1
EWC	59.6±0.6	77.9±0.9	2.8±1.2
Param. Proj.‡	59.2±0.1	79.5±0.3	1.4±0.3
Joint (UB)	64.0±0.4	82.6±0.1	0.0±0.0

Fixed training setup. All runs share one training recipe and differ only in the method and its knob. Backbone ResNet-50 with a 512-unit projector to a 256-d embedding, batch-norm; batch of 32 images with 40 augmented views each (1280 views per step); random crops of 64×64; 20 epochs per task; Adam, learning rate 10^{-3}, weight decay 10^{-6}, gradient clipping at norm 1, bf16. Split Tiny ImageNet into 20, 50 or 100 tasks of equal class count (10, 4 or 2 classes per task). Reported accuracy is the final Class-IL accuracy of a linear probe (50 epochs, learning rate 10^{-2}) after the last task; forgetting is the average drop from each task's best accuracy.

Replay budget. Every method that replays is held to GYRO's budget: a 2000-image reservoir buffer shared across tasks, of which 8 images are replayed per step next to the 32 live ones. DER has, in addition, its own anchor buffer (below), which is part of the method rather than extra replay for the MMCR term.

Protocol. The search runs in two stages. *Stage 1* screens each method's main knob on a single seed (78974) at all three task counts. The values are the ones the CIFAR-100 one-at-a-time sweeps found best, plus one bracket on the other side; for methods with an established default (EWC, CaSSLe, Param. Proj., GYRO) the default cell is the Table 2 run on the same seed, so it is not rerun. *Stage 2* takes the best value per (method, task count) and adds seeds 45466 and 3345, giving the three-seed mean reported in Table S13. A default is kept unless a screened value beats it by more than about 0.5 points on seed 78974; a gap that small is within seed noise (Table 2 std is 0.1–1.1). If the winning value sat at the edge of the screened range, the range was extended by one more value (LUMP α=0.3). Where more than one setting of a method ended up with all three seeds, the one with the higher three-seed mean is reported; single-seed winners often regress (the GYRO and LUMP cells below lost 1–2 points on the two new seeds), and this rule is applied identically to every method. The value can therefore differ across task counts. There is no held-out validation split: selection is on the test accuracy of seed 78974, so the tuned cells are best-of-sweep and mildly optimistic, for the baselines and for GYRO alike.

Search space. Table S11 lists, per method, the knob, what it controls, its default, the values screened and the value chosen at each task count. Every other hyperparameter of a method stays at its code default, listed in the last column.

Outcome. On Tiny ImageNet the CIFAR-100 optima do not carry over. Weak regularisation, which was best on CIFAR-100, is no better than fine-tuning here: EWC needs its default λ=10 at 20 and 50 tasks, and only at 100 tasks, where λ=10 falls below fine-tuning, does a weak λ=0.03 win. CaSSLe is flat in α and keeps its default. Param. Proj. gains 0.6–1.3 points over its default but stays below fine-tuning. NGP prefers rank 8 over rank 1 at every task count. DER is at or below fine-tuning for both α, and DER++, like ER, loses to fine-tuning everywhere. LUMP is the strongest baseline: α=1 beats α=4 at every task count, and α=0.3 won the 20-task screen (29.6) but fell to 28.7±0.7 over three seeds, below α=1 (28.8±0.2). For GYRO, the seed-78974 screen put the default K=10, β=1/3 2–3 points below the best cell at every task count (Figure S1), but over three seeds most of that gap closes: 30.2±1.9 vs 30.0±0.1 at 20 tasks (K=5), 31.2±0.6 vs 31.3±0.9 at 50 (K=10, β=0.5, so the default is kept), and 31.3±0.9 vs 30.6±0.6 at 100 (K=20, β=0.5). GYRO is robust to K and β over $\beta \in [1/3, 0.5]$; $\beta \leq 0.2$ costs several points.

Table S11: Split Tiny ImageNet search space. *Chosen* is the value used in Table S13 at 20 / 50 / 100 tasks.

Method	Knob	What it controls	Default	Screened	Chosen	Held fixed
EWC	λ	weight of the diagonal-Fisher penalty on moving weights important to past tasks	10	0.03, 0.3	10 / 10 / 0.03	Fisher from one full pass over the finished task
CaSSLe	α	weight of the predictive distillation from the live embedding to the frozen previous-task encoder	1	0.01, 0.1	1 / 1 / 1	predictor hidden = 256
Param. Proj.	f	fraction of each weight matrix's top right-singular directions frozen against later updates	0.5	0.01, 0.1	0.01 / 0.1 / 0.1	biases and BN not projected
NGP	rank	output directions per tensor stored per task and projected out of later gradients	32	1, 8	8 / 8 / 8	scale 1; 20 batches of 16 per subspace
DER	α	weight of the penalty on drift of buffered images' embeddings from the ones stored at insertion	1	1, 3	1 / 1 / 3	own ring buffer 2000, 200 inserted per task, 32 drawn per step
DER++	α	as DER, plus the shared replay entering the MMCR term	1	1	1 / 1 / 1	DER buffer as above + shared replay 2000/8
LUMP	α	mixup $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$ between live and replayed views; loss on the mixed views only	0.1	0.3, 1, 4	1 / 1 / 1	8 of the 32 live images mixed per step
GYRO	K	prototypes per task frame	10	5, 10, 20	5 / 10 / 20	frame per task, $D = 256$
	β	weight of the anchor in the mixed centroid, $(1-\beta)c + \beta a$, on current and replayed images	1/3	0, 0.2, 1/3, 0.5	1/3 / 1/3 / 0.5	replay 2000/8

Table S12: Split Tiny ImageNet hyperparameter screen, stage 1 (seed 78974). Final Class-IL accuracy (%) with average forgetting in parentheses. Bold: best value of the method at that task count.

Method	Setting	20 tasks	50 tasks	100 tasks
Fine-tune		26.9 (10.4)	26.0 (8.0)	26.2 (7.0)
EWC	$\lambda = 10$ (default)	**29.1** (9.4)	**27.3** (7.4)	23.7 (9.6)
	$\lambda = 0.3$	26.3 (10.8)	26.4 (10.5)	24.1 (8.5)
	$\lambda = 0.03$	26.7 (9.9)	26.5 (9.9)	**24.8** (7.6)
CaSSLe	$\alpha = 1$ (default)	27.1 (9.7)	**27.4** (7.6)	25.3 (7.3)
	$\alpha = 0.1$	**27.2** (10.1)	26.4 (8.1)	**25.4** (7.2)
	$\alpha = 0.01$	26.7 (10.2)	26.6 (7.7)	25.0 (7.2)
Param. Proj.	$f = 0.5$ (default)	25.5 (9.1)	24.9 (5.4)	22.7 (5.0)
	$f = 0.1$	23.6 (10.7)	**25.5** (7.0)	**24.0** (5.0)
	$f = 0.01$	**26.6** (10.1)	24.1 (6.7)	23.7 (5.3)
NGP	rank 1	25.6 (9.7)	25.8 (7.0)	24.9 (7.1)
	rank 8	**26.7** (10.1)	**27.2** (6.7)	**25.7** (6.4)
DER	$\alpha = 1$	**26.7** (10.1)	**25.7** (8.9)	24.2 (6.6)
	$\alpha = 3$	26.4 (10.2)	24.6 (7.8)	**25.1** (7.2)
DER++	$\alpha = 1$	23.1 (10.4)	22.0 (8.6)	19.4 (6.2)
LUMP	$\alpha = 0.3$	**29.6** (7.9)	26.5 (6.1)	**28.9** (5.6)
	$\alpha = 1$	28.9 (9.5)	**27.5** (6.9)	28.9 (5.0)
	$\alpha = 4$	27.1 (9.5)	25.2 (7.6)	27.7 (5.9)
GYRO	$K = 10, \beta = 1/3$ (default)	29.8 (3.0)	30.2 (3.8)	29.7 (4.3)
	$K = 5, \beta = 1/3$	**32.7** (5.3)	30.6 (3.8)	30.8 (3.9)
	$K = 10, \beta = 0.5$	32.4 (4.0)	**32.0** (3.7)	29.9 (3.7)
	$K = 20, \beta = 0.5$	30.3 (4.8)	30.2 (3.5)	**32.3** (4.4)

Table S13: Split Tiny ImageNet at the tuned setting (stage 2). Final Class-IL accuracy (%), mean±std over seeds 78974, 45466, 3345; the setting used is printed after each cell. A superscript gives the seed count where fewer than three have finished.

Method	20 tasks	50 tasks	100 tasks
Fine-tune	26.0±0.8	25.1±0.7	25.4±0.6
EWC	30.1±0.7 λ=10	27.7±0.3 λ=10	24.6±0.1 λ=0.03
CaSSLe	26.6±0.4 α=1	26.1±1.0 α=1	25.2±0.5 α=1
Param. Proj.	25.9±0.7 f=0.01	25.1±0.4 f=0.1	23.7±0.6 f=0.1
NGP	26.4±0.4 rank 8	27.2±0.6 rank 8	25.5±0.4 rank 8
DER	26.3±0.4 α=1	26.3±0.6 α=1	25.0±0.5 α=3
DER++	21.7±1.4 α=1	21.3±0.5 α=1	19.9±0.6 α=1
LUMP	28.8±0.2 α=1	27.4±0.7 α=1	27.5±1.0 α=1
GYRO (default K=10, β=1/3)	30.0±0.1	31.3±0.9	30.6±0.6
GYRO (best K, β per task count)	30.2±1.9 K=5, β=1/3	31.3±0.9 K=10, β=1/3	31.3±0.9 K=20, β=0.5

GYRO K and β. K is the number of prototypes in each task's frame (a fixed constant, not the class count: every task gets K prototypes whatever it holds). β is the weight of the anchor a in the mixed centroid $(1-\beta)c+\beta a$, applied to both current and replayed images (Section 4.2). β=0 removes the target altogether (and with it any effect of K); β=1 is excluded because the merged row would carry no encoder signal. The Tiny ImageNet screen in the Table 2 setting covers $K \in \{5, 10, 20\}$ and $\beta \in \{0, 0.2, 1/3, 0.5\}$ (β=0 at K=10 only) at all three task counts, seed 78974 (Figure S1). The range was set from the earlier Tiny ImageNet grid at batch 128 × 10 views (Table S14), where $K \geq 20$ with $\beta \geq 0.5$ collapsed. In the Table 2 setting the picture differs: β=0 is the worst column (20.6–23.9) and accuracy rises with β up to 1/3–0.5; the best cell is K=5, β=1/3 at 20 tasks, K=10, β=0.5 at 50 and K=20, β=0.5 at 100. These three cells got seeds 45466 and 3345 (Table S13); with three seeds the default is kept at 50 tasks, and K=5, β=1/3 and K=20, β=0.5 are used at 20 and 100.

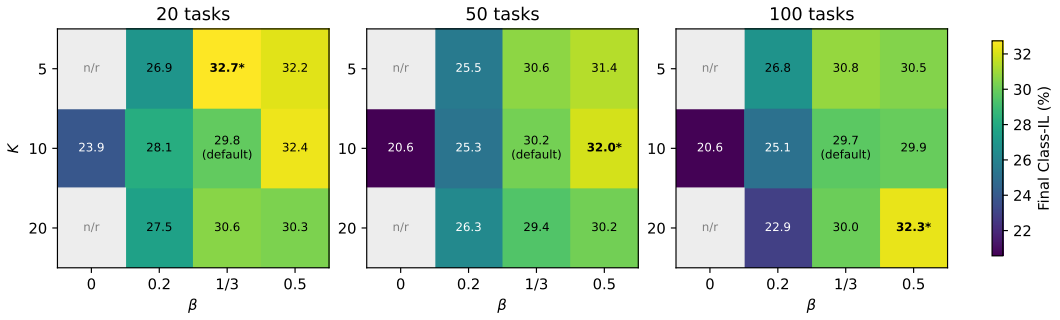

Figure S1: GYRO $K \times \beta$ on Split Tiny ImageNet in the Table 2 setting (ResNet-50, batch 32 × 40 views), seed 78974. Final Class-IL accuracy (%); $*$ marks the best cell per task count; n/r: not run (K has no effect at β=0). The default K=10, β=1/3 cells are the Table 2 runs.

Table S14: GYRO on Split Tiny ImageNet, 100 tasks, ResNet-50, batch 128 × 10 views, replay 2000/8. Final Class-IL accuracy (%), one seed (78974). Bold: best cell. Rows K, columns β.

K	β=0.00	β=0.10	β=0.20	β=1/3	β=0.50	β=0.67	β=0.80
2	**27.2**	26.6	26.5	26.2	26.4	25.5	26.7
5	**27.2**	25.1	25.9	26.4	25.7	26.2	26.3
10	**27.2**	26.1	26.4	26.2	26.5	26.3	25.7
20	**27.2**	26.5	26.2	25.6	24.7	24.5	23.9
40	**27.2**	25.6	25.6	25.7	22.4	22.8	22.1
80	26.5	—	—	—	—	—	—

Table S15: GYRO on Split CIFAR-100, 50 tasks, ResNet-18. Final Class-IL accuracy (%), mean over 3 seeds (a superscript gives the count where fewer finished; —: no finished run). Bold: best cell with all seeds. Rows K, columns β.

K	β=0.00	β=0.10	β=0.20	β=0.50	β=0.67	β=0.80
2	45.9	46.0	49.0	**51.9**	50.2	51.4^{2}
5	45.7	46.3	48.4	51.7	51.8^{2}	—
20	45.7	45.7	46.9	51.4	51.9^{2}	52.5^{1}
40	45.9	46.5	47.2	49.8	50.2	—
80	45.5	46.5	47.1	49.1	50.6	50.4^{1}

H HYPERPARAMETER SWEEPS ON SPLIT CIFAR-100

This section documents the hyperparameter searches behind the Split CIFAR-100 numbers of Table 1, in the same form as the Split Tiny ImageNet searches of Section G.

Fixed training setup. All runs share the recipe of Table S2: ResNet-50 with a 512-unit projector to a 128-d embedding; batch of 32 images with 40 augmented views each; random crops of 32×32; 20 epochs per task; Adam, learning rate 10^{-3}, weight decay 10^{-6}, gradient clipping at norm 1. Split CIFAR-100 is divided into 5, 20 or 50 tasks of 20, 5 or 2 classes. Reported accuracy is the final Class-IL accuracy of the linear probe after the last task.

Protocol. Unlike the two-stage Tiny ImageNet search, every value on CIFAR-100 was run with all three seeds (78974, 45466, 3345), so each cell below is a three-seed mean. Each method's knobs are varied one at a time with everything else at its default; the default cell is the same set of runs as the untuned result, not a rerun. The value reported in Table 1 is the one with the highest three-seed mean among cells where all three seeds finished; where two values tie at the printed precision (NGP at 20 tasks, DER at 5 tasks, GYRO at 20 tasks), bold marks the one used in Table 1. Every replaying method is held to GYRO's budget of 8 replayed images per step from a 2000-image buffer; larger budgets are reported separately (Tables S7 and S21). There is no held-out validation split, so selection is on test accuracy and the tuned cells are best-of-sweep, for the baselines and for GYRO alike.

Table S16: Split CIFAR-100 search space. *Chosen* is the value used in Table 1 at 5 / 20 / 50 tasks; "default" means no swept value beat it.

Method	Knob	Default	Values swept	Chosen
EWC	λ	10	0.01, 0.03, 0.1, 0.3, 1, 3, 10, 100, 1000	0.3 / 0.03 / 0.03
	Fisher batches	whole task	2, 8, 24, whole task	default
CaSSLe	α	1	0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100	0.03 / 0.01 / 100
	predictor hidden	128	32, 128, 512, 2048	default
Param. Proj.	f	0.5	0.01, 0.02, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.99	0.01 / 0.01 / 0.01
NGP	rank	32	1, 2, 4, 8, 16, 32, 64, 128, 256	1 / 8 / 2
	scale	1	0.25, 0.5, 1, 2	default
DER	α	1	0.01, 0.03, 0.1, 0.3, 1, 2, 3, 10, 30	3 / 3 / 10
	own buffer	2000	500, 1000, 2000, 5000	default
ER	replay/step	8	4, 8, 16, 32	8 (budget)
	buffer	2000	500, 1000, 2000, 5000	default
DER++	replay/step	8	4, 8, 16, 32	8 (budget)
	buffer	2000	500, 1000, 2000, 5000	default
LUMP (8/step)	α	0.1	0.1, 0.4, 1	0.4 / 1 / 0.4
LUMP (32/step)	α	0.1	0.02, 0.05, 0.1, 0.4, 1, 2, 4, 8, 16	2 / 8 / 16
GYRO	K (β=1/3)	10	2, 3, 5, 7, 10, 15, 20, 30, 40, 80, 128	2 / 20 / 5
	$K \times \beta$	—	$K \in \{2, 5, 10, 20, 40, 80\}$, $\beta \in \{0, 0.1, 0.2, 0.5, 0.67, 0.8, 0.99\}$	Table S22
	replay/step	8	4, 8, 16, 32	8 (budget)
	buffer	2000	500, 1000, 2000, 5000	default

Outcome. The regularization and distillation baselines are best at their weakest settings, which are the ones closest to fine-tuning: EWC at $\lambda \in [0.03, 0.3]$, Param. Proj. at f=0.01, NGP at rank 1–8 and DER at $\alpha \leq 3$ at 5 and 20 tasks (Table S17). Strong settings cost several points (EWC

Table S17: Split CIFAR-100 sensitivity of the regularization and distillation baselines to their main knob. Final Class-IL accuracy (%), mean±std over seeds 78974, 45466 and 3345. Every other hyperparameter is at its default. Bold: the value used in Table 1. A superscript gives the seed count where fewer than three finished.

Method	Setting	5 tasks	20 tasks	50 tasks
EWC	$\lambda = 0.01$	61.2±0.6	53.6±0.3	49.3±1.4[2]
	$\lambda = 0.03$	61.3±0.4	**54.3±0.4**	**49.6±0.7**
	$\lambda = 0.1$	61.0±0.3	53.8±0.5	48.9±0.3
	$\lambda = 0.3$	**61.5±0.4**	54.0±0.6	47.3±0.9
	$\lambda = 1$	61.2±0.3	53.3±0.6	45.5±0.7
	$\lambda = 3$	61.3±0.3	52.2±0.7	46.1±0.6
	$\lambda = 10$ (default)	60.7±0.6	51.4±0.6	48.6±0.2
	$\lambda = 100$	59.9±0.3	53.2±0.9	48.7±0.2
	$\lambda = 1000$	58.8±0.5	50.8±0.5	45.4±0.0
CaSSLe	$\alpha = 0.01$	61.1±0.4	**54.3±0.6**	49.7±0.1
	$\alpha = 0.03$	**61.4±0.6**	54.1±0.5	49.7±0.6
	$\alpha = 0.1$	60.9±0.2	53.9±0.3	49.7±0.2
	$\alpha = 0.3$	60.8±0.3	54.0±0.6	49.9±0.4
	$\alpha = 1$ (default)	61.4±0.6	53.7±0.5	49.4±0.4
	$\alpha = 3$	61.3±0.4	54.2±0.8	49.6±0.8
	$\alpha = 10$	61.2±0.3	51.9±0.7	48.9±0.2
	$\alpha = 30$	61.0±0.1	51.6±0.5	48.2±0.3
	$\alpha = 100$	61.3±0.7	53.0±0.6	**51.1±0.8**
Param. Proj.	$f = 0.01$	**61.5±0.5**	**54.2±0.3**	**49.8±0.2**
	$f = 0.02$	61.1±0.4	53.3±0.4	49.2[1]
	$f = 0.05$	60.7±0.3	53.2±0.4	49.0±0.3[2]
	$f = 0.1$	60.8±0.7	52.7±0.2	48.0±0.3
	$f = 0.25$	60.1±0.2	52.2±0.9	47.4±0.5
	$f = 0.5$ (default)	59.9±0.6	51.6±0.1	47.1±0.3
	$f = 0.75$	58.4±0.5	51.5±0.4	46.3±0.5
	$f = 0.9$	57.6±0.3	51.0±0.5	46.3±0.4
	$f = 0.99$	55.6±0.2	49.4±0.9	45.3±0.5
NGP	rank $= 1$	**61.3±0.4**	54.3±0.4	50.2±0.4
	rank $= 2$	61.2±0.2	54.2±0.3	**50.2±0.3**
	rank $= 4$	61.0±0.1	54.6±0.4	49.7±0.4
	rank $= 8$	61.2±0.4	**54.6±0.9**	49.3±1.0
	rank $= 16$	61.1±0.1	53.9±0.7	48.7±0.4
	rank $= 32$ (default)	61.1±0.2	53.6±0.2	48.0±0.4
	rank $= 64$	60.4±0.3	52.6±0.1	47.3±0.3
	rank $= 128$	60.0±0.4	52.0±0.3	46.0±0.2
	rank $= 256$	59.5±0.1	52.3±0.1	45.9±0.3
DER	$\alpha = 0.01$	60.1±0.5	53.3±0.5	49.1±0.6
	$\alpha = 0.03$	60.4±0.4	53.1±0.8	48.8±0.5
	$\alpha = 0.1$	60.2±0.2	53.4±0.5	48.9±0.4
	$\alpha = 0.3$	60.3±0.6	53.1±0.6	48.8±0.6
	$\alpha = 1$ (default)	60.3±0.4	53.6±0.7	49.2±0.4
	$\alpha = 2$	60.2±0.5	53.4±0.6	49.2±0.3
	$\alpha = 3$	**60.4±0.4**	**54.0±0.6**	49.8±0.4
	$\alpha = 10$	59.5±1.4	51.8±3.8	**50.1±0.4**
	$\alpha = 30$	55.9±0.6	49.2±1.5	41.9±6.7

Table S18: Split CIFAR-100 sensitivity of LUMP and GYRO to their main knob. Final Class-IL accuracy (%), mean±std over seeds 78974, 45466 and 3345. Every other hyperparameter is at its default. Bold: the value used in Table 1 (for LUMP at 32 replayed images per step, the value used in Table S7).

Method	Setting	5 tasks	20 tasks	50 tasks
LUMP (8/step)	$\alpha = 0.1$ (default)	62.7±0.7	57.6±0.0	54.2±0.3
	$\alpha = 0.4$	**63.2±0.3**	57.9±0.4	**55.2±0.1**
	$\alpha = 1$	62.9±0.2	**58.3±0.6**	54.9±0.5
LUMP (32/step)	$\alpha = 0.02$	60.9±0.3	56.9±0.5	53.4±0.9
	$\alpha = 0.05$	61.4±0.2	57.4±0.3	54.2±0.8
	$\alpha = 0.1$ (default)	62.0±0.5	58.8±0.3	56.2±0.6
	$\alpha = 0.4$	63.1±0.8	60.5±0.4	58.7±0.8
	$\alpha = 1$	63.7±0.3	61.1±0.6	60.2±0.4
	$\alpha = 2$	**63.8±0.4**	61.6±0.5	61.2±0.9
	$\alpha = 4$	63.3±0.5	62.6±0.5	61.2±0.6
	$\alpha = 8$	63.3±0.2	**62.7±0.1**	61.6±0.4
	$\alpha = 16$	62.7±0.1	62.4±0.1	**61.8±0.4**
GYRO (β=1/3)	$K = 2$	**61.7±0.3**	58.7±0.9	56.8±0.5
	$K = 3$	61.5±0.4	58.7±1.1	56.9±0.4
	$K = 5$	61.3±0.5	59.1±0.5	**57.2±0.5**
	$K = 7$	60.9±0.8	58.5±0.6	57.1±0.1
	$K = 10$ (default)	60.9±0.2	59.2±0.2	56.7±0.4
	$K = 15$	60.6±0.5	59.3±0.7	56.8±0.6
	$K = 20$	60.4±0.5	**59.3±0.5**	56.6±0.3
	$K = 30$	60.1±0.6	58.9±0.5	56.9±0.1
	$K = 40$	59.9±0.2	58.9±0.5	56.8±0.4
	$K = 80$	60.2±0.2	58.6±0.6	56.2±0.8
	$K = 128$	59.5±0.6	58.4±0.4	55.6±0.5

Table S19: Split CIFAR-100, default against tuned setting. Final Class-IL accuracy (%), mean±std over seeds 78974, 45466, 3345; the tuned setting is printed after each cell, and the tuned rows are those of Table 1. Methods without a swept knob at the matched replay budget appear once. A superscript gives the seed count where fewer than three finished.

Method		5 tasks	20 tasks	50 tasks
Fine-tune		61.0±0.4	54.4±0.5	50.0±0.7^{2}
ER		60.9±0.4	56.6±0.7	53.6±0.2
DER++		60.3±0.1	55.9±0.4	54.0±0.2
EWC	default	60.7±0.6	51.4±0.6	48.6±0.2
	tuned	61.5±0.4 λ=0.3	54.3±0.4 λ=0.03	49.6±0.7 λ=0.03
CaSSLe	default	61.4±0.6	53.7±0.5	49.4±0.4
	tuned	61.4±0.6 α=0.03	54.3±0.6 α=0.01	51.1±0.8 α=100
Param. Proj.	default	59.9±0.6	51.6±0.1	47.1±0.3
	tuned	61.5±0.5 f=0.01	54.2±0.3 f=0.01	49.8±0.2 f=0.01
NGP	default	61.1±0.2	53.6±0.2	48.0±0.4
	tuned	61.3±0.4 rank 1	54.6±0.9 rank 8	50.2±0.3 rank 2
DER	default	60.3±0.4	53.6±0.7	49.2±0.4
	tuned	60.4±0.4 α=3	54.0±0.6 α=3	50.1±0.4 α=10
LUMP (8/step)	default	62.7±0.7	57.6±0.0	54.2±0.3
	tuned	63.2±0.3 α=0.4	58.3±0.6 α=1	55.2±0.1 α=0.4
GYRO	default	60.9±0.2	59.2±0.2	56.7±0.4
	tuned	61.7±0.3 K=2	59.3±0.5 K=20	57.2±0.5 K=5

Table S20: Split CIFAR-100, second knob of each regularization baseline, main knob at its default. Final Class-IL accuracy (%), mean±std over three seeds. div.: every seed diverged.

Method	Setting	5 tasks	20 tasks	50 tasks
EWC	Fisher batches = 0 (default)	60.7±0.6	51.4±0.6	48.6±0.2
	Fisher batches = 2	61.0±0.7	51.9±0.5	48.2±0.7
	Fisher batches = 8	60.4±0.4	51.4±0.9	48.1±0.5
	Fisher batches = 24	60.9±0.5	51.3±0.4	48.2±0.6
CaSSLe	pred. hidden = 0 (default)	61.4±0.6	53.7±0.5	49.4±0.4
	pred. hidden = 32	61.4±0.6	53.8±0.5	49.6±0.4
	pred. hidden = 512	61.4±0.6	53.9±0.3	49.7±0.5
	pred. hidden = 2048	61.4±0.6	54.2±0.3	49.8±0.4
DER	buffer = 500	60.3±0.2	53.0±0.4	49.3±0.2
	buffer = 1000	60.3±0.4	53.5±0.6	49.2±0.3
	buffer = 2000 (default)	60.3±0.4	53.6±0.7	49.2±0.4
	buffer = 5000	60.3±0.4	53.5±0.5	49.4±0.6
NGP	scale = 0.25	61.2±0.5	54.3±0.2	49.3±0.4
	scale = 0.5	61.2±0.4	54.2±0.3	49.6±0.1
	scale = 1 (default)	61.1±0.2	53.6±0.2	48.0±0.4
	scale = 2	div.	div.	div.

Table S21: Split CIFAR-100, replay budget of the four replay-based methods: images replayed per step (buffer 2000) and buffer size (8 replayed per step). All other settings at their defaults; LUMP at α=0.1, GYRO at K=10, β=1/3. Final Class-IL accuracy (%), mean±std over three seeds; a superscript gives the seed count where fewer finished; div.: every seed diverged.

Method	Setting	5 tasks	20 tasks	50 tasks
ER	replay/step = 4	59.8±0.8	53.8±0.4	50.7±1.0
	replay/step = 8 (default)	60.9±0.4	56.6±0.7	53.6±0.2
	replay/step = 16	62.0±0.4	57.5±0.1	55.7±0.1
	replay/step = 32	60.6±0.3	56.2±0.5	53.8±0.3
	buffer = 500	60.6±0.4	55.5±0.4	52.8±0.6
	buffer = 1000	61.0±0.8	56.4±0.1	53.6±0.7
	buffer = 2000 (default)	60.9±0.4	56.6±0.7	53.6±0.2
	buffer = 5000	61.1±0.6	56.5±0.6	53.4±0.6
DER++	replay/step = 4	58.5±0.5	53.5±0.3	51.5±0.7
	replay/step = 8 (default)	60.3±0.1	55.9±0.4	54.0±0.2
	replay/step = 16	61.3±0.4	57.0±0.2	55.4±0.2
	replay/step = 32	60.2±0.2	56.2±0.3	53.4±0.8
	buffer = 500	60.5±0.3	55.2±0.8	52.3±0.5
	buffer = 1000	60.5±0.2	55.9±0.7	53.0±0.1
	buffer = 2000 (default)	60.3±0.1	55.9±0.4	54.0±0.2
	buffer = 5000	60.6±0.4	55.8±0.6	53.4±0.1
LUMP (α=0.1)	replay/step = 4	62.0±0.5	55.9±0.4	52.1±0.8
	replay/step = 8	62.7±0.7	57.6±0.0	54.2±0.3
	replay/step = 16	62.6±0.3	59.0±0.9	56.4±0.1
	replay/step = 32 (default)	62.0±0.5	58.8±0.3	56.2±0.6
	buffer = 500	60.5±0.1	56.5±0.6	54.9±1.4
	buffer = 1000	61.2±0.4	57.7±0.3	55.0±1.0
	buffer = 2000 (default)	62.0±0.5	58.8±0.3	56.2±0.6
	buffer = 5000	62.8±0.4	60.1±0.6	59.3±0.6
GYRO	replay/step = 4	59.4±0.7	57.3±0.5	54.0±0.9
	replay/step = 8 (default)	60.9±0.2	59.2±0.2	56.7±0.4
	replay/step = 16	61.1±0.1	59.3[1]	div.
	replay/step = 32	59.9±0.4	59.1[1]	div.
	buffer = 500	59.8±0.1	57.2[1]	div.
	buffer = 1000	60.1±0.6	58.1±0.2	54.7±0.1[2]
	buffer = 2000 (default)	60.9±0.2	59.2±0.2	56.7±0.4
	buffer = 5000	61.0±0.2	61.2±0.5	58.9±0.6

λ=1000: 58.8 / 50.8 / 45.4; Param. Proj. f=0.99: 55.6 / 49.4 / 45.3; DER α=30: 55.9 / 49.2 / 41.9). Even tuned, none of them moves more than 1.1 points from fine-tuning (61.0 / 54.4 / 50.0); the largest gain is CaSSLe at α=100 on 50 tasks. Their second knobs change accuracy by at most 0.7 points, except NGP's gradient scale: lowering it to 0.5 gains 1.6 points at 50 tasks, and raising it to 2 diverges (Table S20). Tuning moves GYRO by at most 0.8 points (Table S19): at 20 and 50 tasks every K from 2 to 40 lies within 0.8 points, while at 5 tasks accuracy falls with K, from 61.7 at K=2 to 59.5 at K=128 (Table S18).

Replay budget. Replaying more images per step helps ER and DER++ up to 16 (ER: 60.9 / 56.6 / 53.6 at 8, 62.0 / 57.5 / 55.7 at 16) and drops again at 32 (Table S21). Growing the buffer from 2000 to 5000 changes ER and DER++ by at most 0.6 points but raises LUMP (62.0 / 58.8 / 56.2 to 62.8 / 60.1 / 59.3) and GYRO (60.9 / 59.2 / 56.7 to 61.0 / 61.2 / 58.9). With the whole batch mixed, LUMP peaks at α=2 on 5 tasks and α=8 on 20 tasks, and is still rising at α=16 on 50 tasks, so its optimum there is not bracketed. These larger budgets are excluded from Table 1, which holds every method at 8 replayed images per step.

GYRO K and β. Table S22 crosses K with β; Figure 3 plots the same grid as the gain over β=0. At 20 and 50 tasks every cell with $0 < \beta \leq 0.8$ is at or above β=0. At 5 tasks large frames combined with strong anchoring fall below it (K=80, β=0.8: 57.3 against 60.8). β=0.99 is the worst column wherever it was run, since the merged row then carries almost no encoder signal. The best cell with all three seeds is K=2, β=0.8 at 5 tasks (61.9), K=5, β=0.2 at 20 tasks (59.0) and K=5, β=0.5 at 50 tasks (57.4). The grid does not include β=1/3, the value used in Table 1, where only K is tuned (Table S18).

Table S22: GYRO $K \times \beta$ grid on Split CIFAR-100, ResNet-50, same training setup as Table 1. Final Class-IL accuracy (%), mean over seeds 78974, 45466 and 3345; a superscript gives the seed count where fewer finished; —: no finished run. β=0 applies no merge, so one set of runs serves every K. Bold: best cell with all three seeds. Figure 3 shows the same grid as the gain over β=0.

K	β=0	β=0.10	β=0.20	β=0.50	β=0.67	β=0.80	β=0.99
5 tasks							
2	60.8	61.3	61.7	61.3	61.5	**61.9**	55.2
5	60.8	61.1	61.2	60.9	60.7	60.8	53.7
10	60.8	60.9	61.2	60.5	60.0	59.4	53.5
20	60.8	61.0	60.8	59.7	58.6	58.7	52.6
40	60.8	61.1	60.3	59.4	58.4	57.9	52.0
80	60.8	60.8	60.3	58.5	58.2	57.3	51.6
20 tasks							
2	56.6	57.8	58.6	58.2	58.8	58.6	56.1
5	56.6	57.2	**59.0**	58.7	58.8	58.9^{2}	55.7
10	56.6	57.5	58.6	59.0	58.3^{2}	58.1^{1}	55.3^{2}
20	56.6	57.3	58.6	58.5	58.3^{2}	—	54.6
40	56.6	57.1	57.6	58.2	58.3	57.8^{2}	55.9^{1}
80	56.6	56.6	57.5	58.2	57.4	57.2^{2}	55.1^{2}
50 tasks							
2	53.7	55.2	57.2	57.1	56.8^{1}	57.6^{1}	—
5	53.7	54.5	56.1	**57.4**	—	57.6^{1}	—
10	53.7	54.6	56.0	56.6^{2}	56.6^{2}	57.2^{1}	—
20	53.7	54.4	55.6	56.3	56.8	56.9^{1}	—
40	53.7	54.8	55.8	56.2	56.6^{2}	56.7^{1}	—
80	53.7	54.2	54.8	56.9	56.6	—	—